

Debreceni Egyetem

Informatikai Kar

'Kisbagoly' – Az OTDK online nevezési rendszere

Témavezető:
dr. Végh János
Egyetemi tanár

Készítette:
Miskolczi Zsolt
Programozó matematikus
szakos hallgató

Debrecen

2009

Tartalomjegyzék

1. Bevezetés	4
1.1. Az előzmények, a szükségesség indoklása	4
1.2. Követelmények „madártávlatból”	6
1.3. Az OTDK bemutatása az adminisztrációs folyamatok szemszögéből	6
2. Az alkalmazás tervezése	7
2.1. Előzmények, az eddigi évek gyakorlata	7
2.2. A program leírása	8
2.3. Követelmények	10
2.4. Fogalomtár	12
2.5. Szerepkörök	14
2.6. Forgatókönyvek	19
2.7. Az adatbázis felépítése	20
2.8. Az architektúra	37
2.9. konkrét példa (szemléltetésként)	39
3. Az alkalmazás áttekintése	60
4. Összefoglaló és köszönetnyilvánítás, jövőkép	63
5. Függelék	65
5.1. A szükséges technológiák és eszközök	
5.2. Használati útmutató	
5.3. Irodalomjegyzék	
5.4. Elismerések	

1. Bevezetés

1.1 Az előzmények, a szükségesség indoklása

A 2007/2008-as tanév őszi félévében indult dr. Végh János vezetésével egy Csoportos szoftverfejlesztés nevű kurzus, melyre én is jelentkeztem. A kurzus során megismerkedhettünk az egyetemmel kapcsolatban álló cégekkel és lehetőségünk volt éles, valós életbeli fejlesztésekbe belekerülni. Tulajdonképpen ezen a kurzuson találkoztam az Országos Tudományos Diákköri Tanács titkárságával, akiknek segítségre volt szükségük a nagy hagyományokra visszatekintő Országos Tudományos Diákköri Konferencia szervezési feladatainak a papír alapú ügyintézésről elektronikussá való formálásában.

Maga a konferencia hosszú évek óta segíti a felsőoktatás hallgatóinak tudományos fejlődését két évente megrendezett országos megmérettetés formájában. Ezen országos versenyre az ország bármely tájáról jelentkezhetnek hallgatók (sőt még a határon túlról és bizonyos kivételes esetekben középiskolákból is) szakterületeikkel kapcsolatosan elkészített tudományos dolgozataikkal, hogy megmérettethessék magukat egy, az egész ország hallgatóit felölelő versenyen. Jelenleg a XXIX. OTDK zajlik, ennek a konferenciának kell segítő kezet nyújtanunk egy jól használható nevezési programmal.

Mivel maga a konferencia már hosszú évekre tekint vissza, megvannak a nevezési folyamatnak a maga kialakult hagyományai. Jellemzően az évek során a nevezés minden mozzanata papír alapon történt, postai kézbesítéssel. A postának több nagy hátránya van, ezek a mai világban már elfogadhatatlanok, és fölösleges idő- és pénzpazarlást jelentenek. Egyik hátrány a sebesség. Egy postai csomag vagy levél kézbesítése szerencsés esetben is több napig zajlik, rossz esetben akár egy hetet is igénybe vehet. Második nagy gond a megbízhatóság. A csomagok és levelek kézbesítésére sajnos jelenleg hazánkban nincs garancia. Nincs garancia arra, hogy a csomag épségben érkezik meg, de még arra sem, hogy egyáltalán megérkezik a végcélhoz. A harmadik probléma a költség. A nagy mennyiségű dokumentum folyamatos postázása egy-egy személy esetében nem lehet túl sok, de ha összeszámolunk mindenkit, már igencsak tetemes összeg jöhet ki a nevezési időszak alatt. Ezen problémák csökkentésére az elmúlt években az internet és az e-mail megjelenítésével voltak már kísérletek. Jellemzően az utóbbi években a kötelezően

postai úton kézbesítendő nevezési dokumentumok mellett a titkárság és az OTDK szekcióinak rendezői között már kialakult egy e-mail formátumban zajló levelezés a rövidebb megbeszélőnivalók (pl.: hiányközlések, átsorolások) meggyorsítására. Maga a nevezés az országos megmérettetésre ettől függetlenül még mindig kizárólag papír alapon zajlott. Előzőleg a XXVIII. TDK-n volt lehetőség elektronikus nevezésre. Ezzel a felülettel sajnos nem volt szerencsém találkozni. Valószínűleg ez egy kezdetleges, egyszerű, elsősorban adatfelvitelt szolgáló webes program volt, és ez hozhatta meg a titkárság kedvét egy szélesebb körben használható alkalmazás felé.

Jellemző volt az előző évek gyakorlatára a szétszóródottság. Sok esetben a szükséges formanyomtatványok már az országos verseny szekciói számára is eltértek. A helyi versenyek rendezői jegyzőkönyveket készítettek az intézményeikben tartott fordulókról, és ezeket postán küldték el a titkárság számára. Ezzel párhuzamosan a pályamunkák szerzői szintén formanyomtatványokat töltöttek ki és postáztak az OTDT titkárság, valamint az OTDK tudományos szekcióinak rendezői számára. A programot megelőző időkben a fő probléma a szekciók rendezői és a titkárság közötti kapcsolattartás formája volt. Jellemzően a hivatalos dokumentumokat postázták, valamint e-mailben tartották a kapcsolatot egymással. A különböző statisztikákat a rendezők elkészítették, elküldték a titkárságnak, és ők „összeollózták” őket. Ezekből az összemásolgatott táblázatokból készültek el az országos versenyek eredményeit közzétevő Diáktudós kiadványok is.

Az előző években nem igazán volt egységes nevezési rendszer. Az egyetlen kezdeményezés, amit ismerek, egy webes alkalmazás volt php alapokon, ez a XXVIII. Országos Tudományos Diákköri Konferenciát szolgálta ki. Ezzel a programmal sajnos nem volt lehetőségem találkozni, így ebből nem tudtam kiindulni az alkalmazás tervezésénél. Az OTDT titkárság anyagi keretei miatt egy egyedi szoftverfejlesztés nyilvánvalóan megfizethetetlen, így mind a titkárság, mind számunkra, hallgatók számára kapóra jött ez a feladat a Csoportos projektmunka című kurzus keretein belül. A program fejlesztését először ketten kezdtük el Szatmári András kollégámmal, később csatlakozott hozzánk Kocsis Dániel is. A feladatok általában a következőképpen alakultak: A program tervezését és a felületek, valamint az üzleti logika és az adatelérés jelentős részét én készítettem, akárcsak az adatbázist is. Szatmári Andrásra jutott a felhasználók és a jogosultságok kezelése, valamint a felhasználói kézikönyv megírása és a felhasználókkal való kapcsolattartás. Kocsis

Dániel kapta meg a nyomtatási feladatokat, melyeket pdf formátumú állományok formájában oldott meg. Ezek a szerepkörök kicsit kicserélődtek, amikor nekikezdttem a szakdolgozat megírásának, ekkor András vette át a szerepemet.

1.2 Követelmények „madártávlatból”

Az előző évek gyakorlatából szinte egyenesen következnek a program alapvető követelményei. Elsődleges cél a postai forgalom minimalizálása, illetve a program esetleges későbbi széles körben való elfogadása után teljes megszüntetése, így mind a kommunikációs csatornák gyorsítása, mind a költségek csökkentése.

További fontos követelmény az adatok helyességének megőrzése és az adatok ellenőrzésének biztosítása. Mivel a felhasználók jelentős része még ma is bizalmatlan az elektronika iránt, számukra világosan láthatóvá kell tenni, hogy egy elektronikus program igenis használható, és igenis meg lehet bízni benne. Ez persze nem mehet egyik napról a másikra, egy hosszú és nehéz folyamat lehet, ami előbb-utóbb meghozhatja a gyümölcsét, és egyszer talán el lehet felejtetni a töménytelen mennyiségű papír folyamatos nyomtatását és küldözgetését. Rendkívüli módon lényeges, hogy megmutassuk a felhasználóknak azt, hogy egy webes program megbízható, és nem kell félni az adatok világgá kürtölésétől, hűtlen kezelésétől. Talán ezt is elősegíti dr. Pataricza András, az Országos Tudományos Diákköri Tanács Informatikai Szakmai Bizottságának elnökének felajánlása. Ő felajánlott egy üveg pezsgőt annak, aki feltöri a rendszert. Tudtommal a pezsgő még a dolgozat írásának az időpontjában is gazdára vár és remélhetőleg belátható időn belül nem is talál gazdára.

További cél volt az alkalmazás szükségességét indokolandó, hogy az országos megmérettetés eredményeiről szóló Diáktudós kiadvány számára is lehessen statisztikákat szolgáltatni. Ezzel rengeteg irodai munkát sikerülne levenni az OTDT titkárságának a válláról, így további időt és pénzt takarítanánk meg.

1.3 Az OTDK bemutatása az adminisztrációs folyamatok szemszögéből

Az Országos Tudományos Diákköri Konferencia két évente kerül megrendezésre. A konferenciára a helyi versenyeken sikeresen részt vevő hallgatók jelentkezhetnek a pályamunkáikkal.

Egy adott pályamunka elkészítésében több hallgató is részt vehet, akiknek konzulensek segítenek a munkában. A hallgatók jelentkezhetnek hazai és külföldi felsőoktatási intézményekből, valamint speciális esetben középiskolából is. A pályamunkához szorosan hozzátartozik egy rezümé is, ami egy rövid összefoglaló a pályamunka tartalmáról.

A konferencia több szekcióban kerül megrendezésre (a 2009-ben megrendezett, XXIX. versenyen 16 szekció került meghirdetésre). A szekciók tagozatokra bomlanak. Ezeken a szekciók rendezői a nevezési időszak alatt is változtathatnak, új tagozatokat vehetnek fel.

A hallgatók egy dolgozatot egy adott szekcióra, és azon belül két tagozatra jelentkeztetnek. Megadhatnak egy „Legmegfelelőbb tagozatot”, és egy „Megfelelő tagozatot”. Ezeket a jelentkezéseket a szekciók rendezői saját hatáskörükön belül szabadon átsorolhatják a különböző tagozatok között. A hallgatóknak a jelentkezést az országos megmérettetésre az elektronikus forma mellett (ezen programon keresztül), papírral is meg kell erősíteniük. Ehhez mind a Titkársághoz, mind a rendezőkhöz el kell juttatniuk a szükséges dokumentumokat. A Titkárságnak és a rendezőknek a programban kell visszajelezniük a dokumentumok beérkezését, és az esetleges hiánypótlások szükségességét. Egy pályamunka nevezése akkor hagyható jóvá, ha mind a Titkárság, mind a szekció rendezője elektronikusan jóváhagyta azt.

2. Az alkalmazás tervezése

2.1 Előzmények, az eddigi évek gyakorlata

Az OTDT titkárság fennállása során elsősorban papír alapú ügyintézkést folytatott. Elektronikusan belső anyagokat tartottak, nem volt egységes kommunikáció köztük és a rendezők, intézményi felelősök, valamint a hallgatók között. A Titkárság a nevezésekről, és az országos verseny eredményeiről szóló statisztikákat belső Excel táblázatokra támaszkodva állította elő, melyeket ők „ollóztak össze” a szekciók rendezői visszajelzései alapján. Ezekre támaszkodva készültek el a számukra szükséges, a vezetők számára szóló adatszolgáltatások is. Így készültek el az eredményekről szóló Diáktudós kiadványok is.

A régebben bevett szokás szerint az intézményeknél tartott helyi versenyekről az összesítő lapok postai kézbesítés útján jutottak el a titkársághoz. A hallgatók a műveikkel szintén levélben jelentkeztek. A visszajelzések a hallgatókhoz, és a rendezőkhöz tipikusan postán illetve kisebb jelentőségű kérdéseknél esetleg e-mailben

valósultak meg. A postával hatalmas probléma a megbízhatatlanság, a hosszú átfutási idő, és a drága költségek. Tekintetbe véve, hogy az idő pénz, már ezért is célszerű volt egy elektronikus nevezési rendszer bevezetése. A nevezésekhez szükséges formanyomtatványok sok esetben szekciónként eltérőek voltak. Ezek kezelése nehézkes lehet, hiszen minden szekcióhoz adott esetben különböző folyamatokat szükséges rendelni a feldolgozásukhoz. Ez is többlet időhöz vezet.

2.2 A program leírása

Maga a program egy webes alkalmazás, mely böngészőben futtatható. Megtekintéséhez támogatott az Internet Explorer böngésző használata, bár Firefox alatt is használható. Néhány kisebb piaci részesedésű böngészővel problémák lehetnek, ilyen például a Google Chrome. A fejlesztés időigénye, és a követelmények gyors változása miatt egyelőre nem tervezett a böngészők közötti széles támogatottság elérése, és az alkalmazás eddig sem volt letesztelve ezeken.

Az idő rövideje és az alapvető, a program lényegét meghatározó elvárások gyors, akár hétről-hétre történő változásának köszönhetően nem igazán volt lehetőség letisztult tervezési módszertan használatára. Ez egyrészt betudható annak, hogy a megrendelő távol áll az informatikusi szakmától, így problémák adódhatnak azzal, hogy nem teljesen azt mondja, amit szeretne. Másrészt nyilvánvalóan saját tapasztalatlanságunk is megmutatta magát. A program készítésekor megszerzett talán egyik legfontosabb tapasztalat az, hogy a legfontosabb tudni, hogy hogyan, és mit kell kérdezni. Mivel a megrendelő nem képes ezt közölni, a programozó feladata kiszedni belőle a szükséges információkat. Sajnos kezdeti tapasztalatlanságainknak köszönhetően dokumentációként a fejlesztés során csak a megbeszélések alatt készült jegyzőkönyveket használtuk, más tervezéshez szükséges anyag nem nagyon született.

A módszerünk talán leginkább a prototípus alapú szoftverfejlesztéshez hasonlítható. A megbeszélések során jegyzőkönyvet készítettem. Ezeket később összegeztem, és az összegezést küldtem el a tagoknak (tipikusan a titkárságnak és a fejlesztésben jelentős segítséget nyújtó Szatmári András kollégámnak). Miután megérkeztek a visszajelzések, újra összegeztem a követelményeket, ezután kezdődött a fejlesztés. Az elkészült munkát megmutattuk a megrendelőknek, és természetesen ilyenkor időnként előkerült a „Mi nem ilyen lovat akartunk” című eset, ekkor minden

kezdődött előről. Ha megfelelt, akkor a jogosultságok beállítása után a felhasználók számára is elérhetővé tettük a változást.

A szoftver jelenleg az alábbi környezetben fut:

- Microsoft Server 2003 Enterprise x64, SP2
- Microsoft SQL server 2005
- IIS6
- .Net framework 3.5 SP1

Tervbe van véve az SQL server frissítése 2008-as verzióra (tapasztalataim szerint jobban teljesít, és még ráadásul újabb indexek is bevezetésre kerültek a 2005-ös verzióhoz képest, szóval mindenképpen célszerű változás), valamint a Reporting Services használata is a statisztikákhoz és az adatszolgáltatásokhoz.

A programnak jelenleg a Debreceni Egyetem ad otthont, elérhető az otdk.unideb.hu címen. Továbbá van egy tartalék szerverünk, ez az otdk.inf.unideb.hu-n található, ez a „véstartalék”. Felhasználóinkkal fenntartunk továbbá egy plusz kommunikációs csatornát e-mail formában, a cím otdkAdmin@gmail.com.

A fejlesztés természetesen folyamatban van ezen dolgozat megírásának időpontjában is, viszont az alkalmazás már használható állapotban elérhető. A program eddigi követelmények alapján várható befejezése 2009 év harmadik negyedére várható. A program eddigi funkcióival általában elégedettek, és bár néhány funkció még hibajavításra szorul, de jelenleg is használható. A használhatóságot talán a több mint 4000 nevezés is bizonyítja, sőt az országos verseny nagy részét már le is bonyolították rajta.

A szoftver egy több rétegű architektúrára épül. A rétegek kifejtésére később kerül sor, mindenesetre az már most elárulható, hogy az adatok elérésére egy Microsoft által kifejlesztett ORM mapping eszközt, a Linq-t és a Linq To Sql osztályokat használjuk. Így az adatbázistábláinkat megfeleltethetjük konkrét C# osztályoknak, nincs szükség SQL utasítások írására. Ezzel a program áttekinthetőbb és logikusabb felépítést kap, mintha lépten-nyomon tárolt eljárásokba futna az ember és mindegyiket kézzel kellene hívogatni a különböző osztályokból. Ezzel természetesen csökken a fejlesztési idő, és átláthatóvá válik a forráskód, ami egy igen fontos tényező,

mivel a változtatások gyakoriak, és a fejlesztési idő viszonylag hosszú mondjuk egy általunk megszokott 1-2 hetes vagy napos és 100 sorral rendelkező egyetemi beadandóhoz képest. A több rétegű architektúra további előnye, hogy a program élesen részekre bontható. Külön kezelendő az adatok elérése és manipulálása, az üzleti logika, és a felhasználói interfész, vagy esetlegesen, ha később megnőne a program, akkor a felhasználói interfészek. Ez a heterogén felépítés is segíti a gyors változások követését. Egy logikai változáshoz nem kell átírni a felületet, és egy megjelenítési módosításhoz nem kell hozzányúlni a mögötte lapuló algoritmusokhoz sem.

Maga a szoftver Microsoft eszközök felhasználásával készült, melyekhez az MSDNAA licensznek köszönhetően értünk el. Elkészítéséhez a Visual Studio 2008 nevű szoftverfejlesztő eszközt használjuk fel, a forrása egy Visual Studio solution-ba van foglalva. A solution tulajdonképpen logikailag összekapcsolódó projektek együttese, a különböző rétegek különböző projektekben szerepelnek.

2.3 Követelmények

A követelmények szorosan az OTDT instrukciói alapján, és velük szorosan együttműködve lettek meghatározva. A szoftver tervezésénél, és az első megbeszéléseknél az alábbi követelmények lettek felvázolva:

- A rendszer minden változást tárol, a változtatást végző személy azonosítójával és a változtatás időpontjával együtt.
- Az adott összefüggésben illetékes személy, vagy szervezet a határidőn belül saját illetékességi körében mindent tud változtatni.
- Kétféle változtatás van: amit az érintett/illetékes maga el tud végezni, és amit a jogosultság/időtúllépés miatt csak a felügyelőktől tud kérni. Az utóbbi esetben elfogadást és elutasítást is indokolni kell.
- Az adott összefüggésben felügyelői jogú illetékes személy, vagy szervezet bármikor tud változtatni, a saját illetékességi körén belül.
- Az illetékességi hatáskör felfelé terjed, pontosan meg kell határozni, hogy milyen tranzakció felügyeletére kik illetékesek.
- A változások tényéről az illetékes felügyelő személyeket/szervezeteket a rendszer levélben értesíti.

- A rendszer országos TDK versenyek egységeiben dolgozik, a lezáráskor automatikus archiválást végez.
- A rendszer lehetőséget biztosít arra, hogy az intézmények helyi TDT elnökei regisztrálják a rendszerben az intézményi/kari konferenciát.
- Az egyes intézmények rekordjában tárolja a verseny időpontját, adatait.
- Az OTDT titkárság visszaigazolásairól, az intézmény rekordján végzett változásokról levélben értesíti a helyi TDT elnököt.
- A rendszer a jegyzőkönyvek adatai alapján listát készít az intézményből az OTDK-ra nevezni jogosult hallgatókról.
- A rendszer lehetőséget biztosít arra, hogy a rendezők nyugtázzák a helyi konferencia dokumentumainak megérkezését.

További követelmények, amiket én támasztottam:

- Az adatok biztonságának biztosítása. A jelszavak titkosított letárolása.
- Elérhetőség. A rendszer bármikor elérhető legyen.
- Megbízhatóság. Jelentős sebességvesztés nélkül működjön a nagy roham idején is (az országos TDK versenyekre való nevezés időszakában)
- Kezelhetőség. A rendszer legyen könnyen és megbízhatóan kezelhető

A követelmények jelentős része már implementálva van, más részeik az év nyara során készülnek el. Ami még hiányosság ezek közül (a hiányosságok pótlása és a nagy átnézés a nyárra van beütemezve):

- A változások tárolása, logolás
- Az adatok automatikus archiválása
- Kezelhetőség. A megrendelő folyamatos módosításainak és a politikának köszönhetően ez sajnos a háttérbe került.

Az e-mail küldése minden egyes változás után később el lett vetve, mivel nem akartuk fölöslegesen leterhelni a felhasználók fiókjait. Ehelyett bevett szokássá vált, hogy körüzeneteket küldünk a felhasználóknak a fontosabb eseményekről.

2.4 Fogalomtár

Az országos versenyeknek, és a levezetésüknek természetesen megvan a saját fogalomtáruk. Ennek a jelen programra vonatkozó részét a megértést elősegítendő igyekeztem összegyűjteni. Ez a fejezet ezt a listát tartalmazza:

Szekció: Az Országos Tudományos Diákköri Konferencia tudományterületeinek osztályainak elnevezése.

Tagozat: A szekciók tudományterületi sajátosságainak megfelelő további bontásaival létrejött al-szekciók elnevezése.

Pályamunka, pályamű: Egy vagy több hallgató által elkészített dolgozat vagy pályamű, amit az intézményenkénti/kari diákköri konferencián bemutattak a szerzők, és ott jogosultságot szereztek az OTDK-n való részvételre.

Rezümé: A pályamunka rövid szöveges összefoglalója.

Szerzők: A pályamunka elkészítésében részt vevő hallgatók.

Intézmény: A felsőoktatási intézmények Magyarországon, valamint a határon túli tudományos diákköri konferenciát rendező intézmények és szervezetek, továbbá a középiskolások azon versenyeinek rendezését végző szervezetek, ahol az OTDK-n való szereplésre jogosultságot lehet szerezni.

Kapcsolattartó: A pályamunka szerzői közül egy hallgató, aki az elektronikus rendszerben regisztrálja és jelentkezteti a pályaművet. A titkárság nem használja, kizárólag a programban jelenik meg.

Konzulens: A pályamunka elkészítésében a hallgatókat segítő oktató, kutató, mestertanár, vagy intézményen kívüli, a pályamunka témájának területén elismert szakember.

Kari felelős: A felsőoktatási intézmény adott karának TDT elnöke, aki felel a kari diákköri konferenciák, valamint az OTDK-ra való nevezés dokumentációjának az OTDT Titkárságára, a nevezési anyagok esetében pedig a titkárság és a szekciót rendező intézmények felé történő megküldéséért.

Intézményi felelős: A felsőoktatási intézmény TDT elnöke, aki felel az intézményeknek az OTDT felé megküldött adatokért és dokumentumokért.

Országos szekció rendezője, rendező: A különböző tudományterületi szekciók ügyvezető elnökei.

OTDT Titkárság: Az Országos Tudományos Diákköri Tanács Titkársága, amely felelős az OTDK központi koordinálásáért és szervezéséért. Továbbiakban **titkárságként** is hivatkozok rá.

OTDT titkársági tag: Az OTDT Titkárságának munkatársai, akik a szervezést koordinálják, és a központi adminisztrációért felelősek.

Helyi jegyzőkönyv: Az intézményi/kari diákköri konferencia lebonyolítása során keletkező dokumentum, amely kötelezően tartalmazza a helyi konferencia időpontját, jellegét és helyszínét, a bemutatott pályamunkák címét, szerzőjének vagy szerzőinek nevét, aktív fél éveinek számát, azt a szakot, amelyet a felsőoktatási intézményben végeznek, a pályamunkák témavezetőinek nevét és beosztását, az értékelést végzők (zsűri/és vagy bírálók) nevét, a pályamunka minősítését és az OTDK-n való részvételre vonatkozó javaslatot.

Az OTDK jegyzőkönyve: az Országos Tudományos Diákköri Konferencia eredményeit tartalmazó dokumentum.

Nevezési dokumentumok: Az Országos Tudományos Diákköri Konferenciára való nevezéshez a postai úton elküldendő dokumentumok a szekciók rendezői és a titkárság számára.

- Rendezők számára:

- Nevezési lap másolata (az elektronikus rendszerből nyomtatva)
- Rezümé nyomtatott formában
- Dolgozat

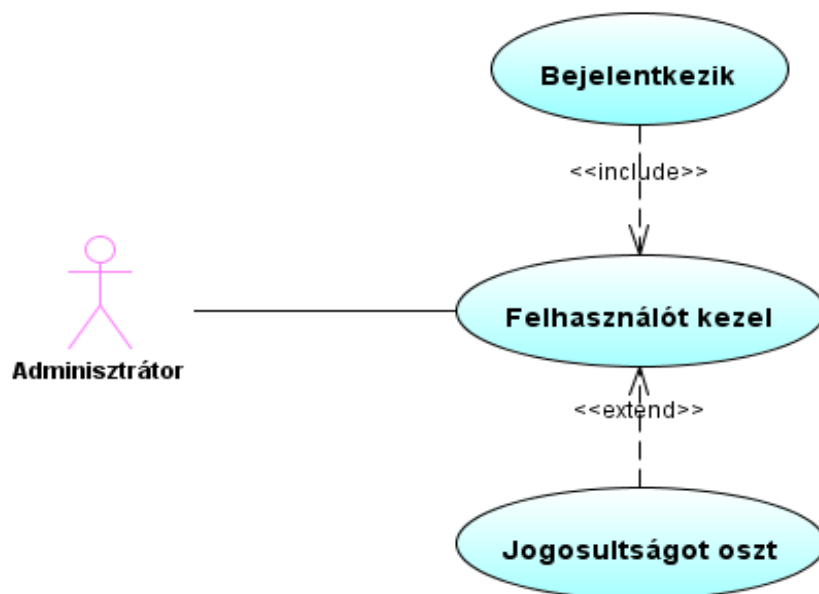
- A titkárság számára:

- Nevezési lap (az elektronikus rendszerből nyomtatva)
- Rezümé nyomtatott formában
- Végzett hallgatók esetében nyilatkozat (ha a végzés félévében szerzett jogosultságot az országos konferencián való részvételre, a nevezési laphoz csatolnia kell témavezetőjével tett közös nyilatkozatát arról, hogy a végzés félévében bemutatott TDK dolgozat egy korábban megkezdett műhelymunka eredménye, s az minden tekintetben megfelel a felhívásban foglalt követelményeknek)

2.5 Szerepkörök

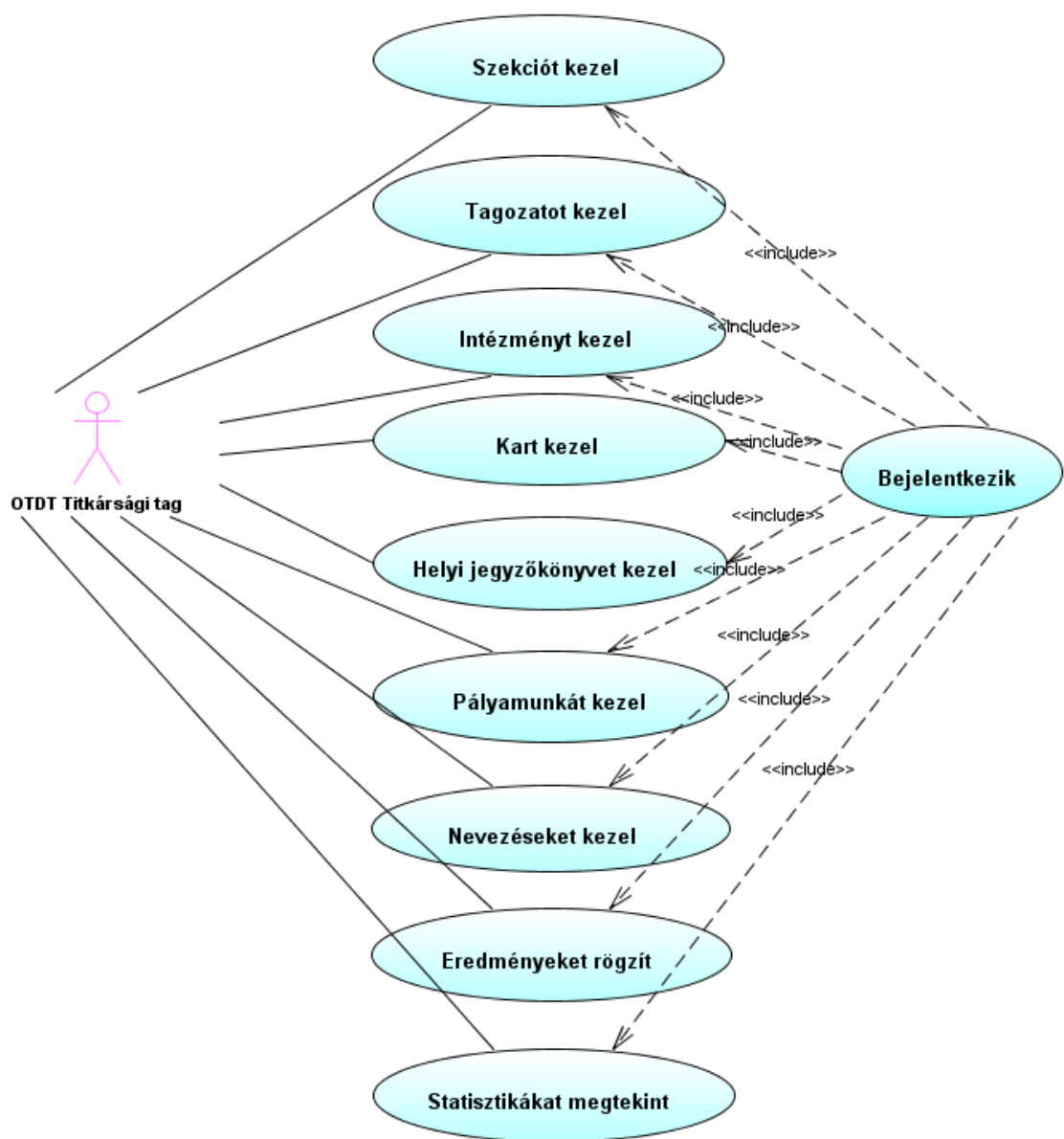
A rendszer az alábbi szerepköröket ismeri: adminisztrátor, OTDK titkársági tag, OTDK bíráló, országos szekció rendezője, helyi TDT elnök, kari TDK felelős, hallgató. A szerepkörök demonstrálására UML diagramokat használunk.

- **Adminisztrátor** (admin): Kezeli a felhasználókat, jogosultságot oszt.



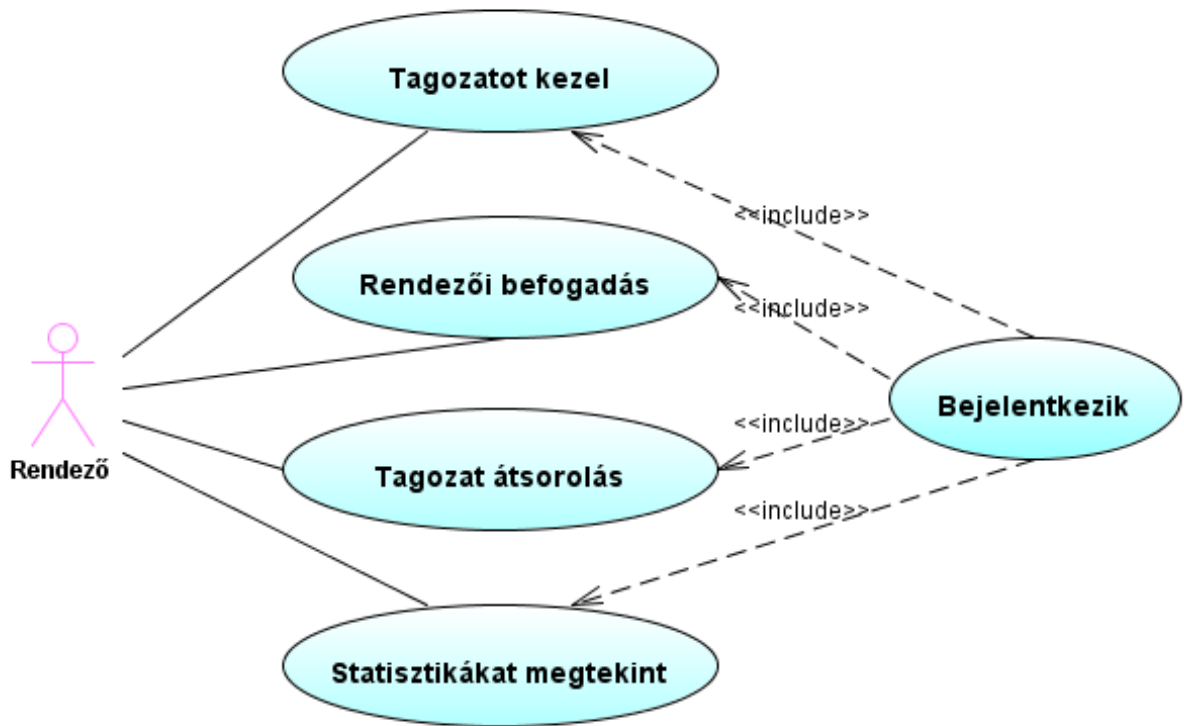
■ 1. ábra: Az adminisztrátori szerepkör bemutatása UML diagram segítségével

- **OTDK titkársági tag:** Karbantartja a szekciókat és tagozatokat, valamint az intézményeket. Módosíthatja a helyi versenyek adatait, valamint új helyi versenyeket vehet fel. Képes dolgozatot fölvenni, nevezni a versenyre, nevezést törölni. Hozzáfér a különböző adatszolgáltatásokhoz, listákat tud lekérni a rendszerből. Visszaigazolja a nevezési dokumentumok beérkezését, jóváhagyhatja a dolgozat nevezését az országos versenyre. Ezt csak abban az esetben teheti meg, ha a rendező is jóváhagyta a nevezést. Törölhet nevezést és dolgozatot, átsorolhat egy nevezést másik szekcióba.



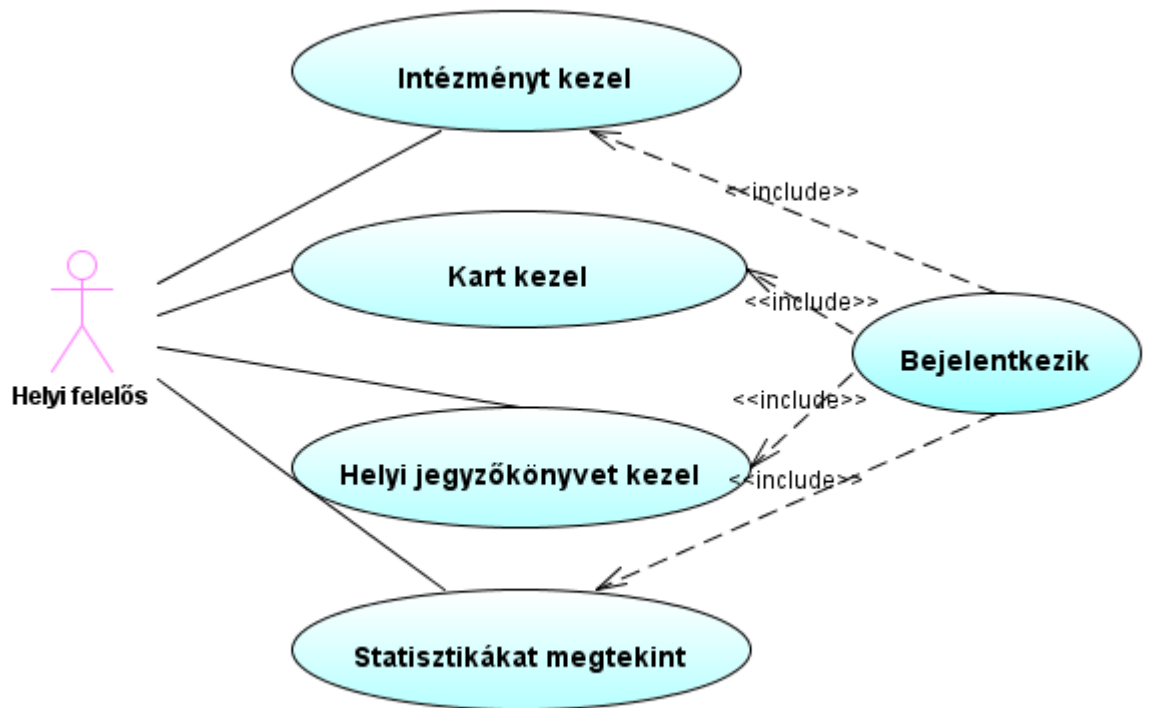
▪ 2. ábra: Az OTDT Titkársági tag szerepkörét reprezentáló UML ábra

- **OTDK bíráló:** jelenleg nem végez műveleteket a rendszerben. Jövőbeli felhasználásra fenntartva.
- **Országos szekció rendezője:** Visszajelzéseket ad a nevezési dokumentumok beérkezéséről. Tagozatokat kezel, új tagozatokat vehet fel, tagozat átsorolást hajthat végre.

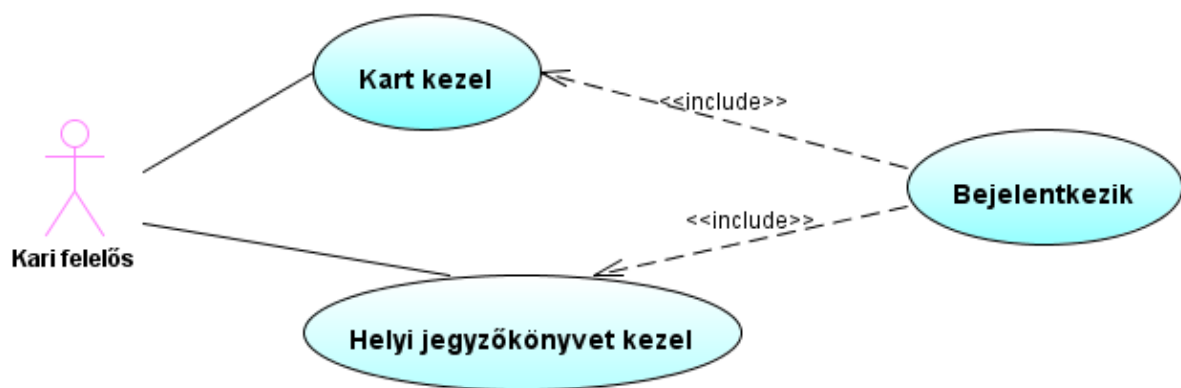


3. ábra: Az országos szekciók rendezőinek szerepköreit bemutató UML diagram bemutatása

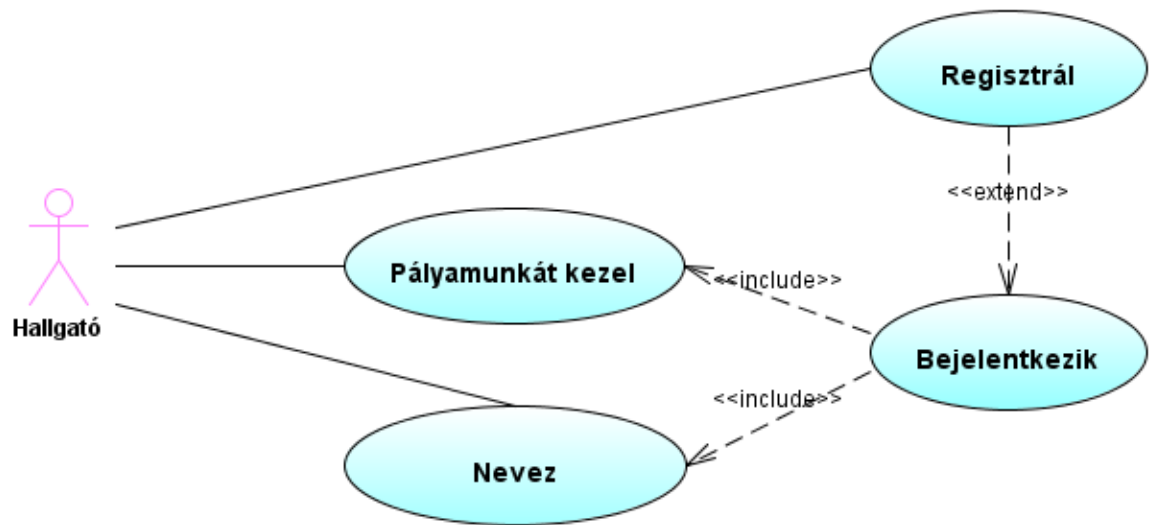
- **Helyi TDT elnök:** Karbantartja az intézménye, és intézményének a karainak az adatait. Kari felelőst nevezhet ki. Felviszi a helyi jegyzőkönyv legfontosabb jellemzőit.
- **Kari TDK felelős:** Változtathatja a karjának tulajdonságait.
- **Hallgató:** Regisztrálhat a rendszerbe. Pályamunkát rögzíthet, illetve nevezhet az országos versenyre.



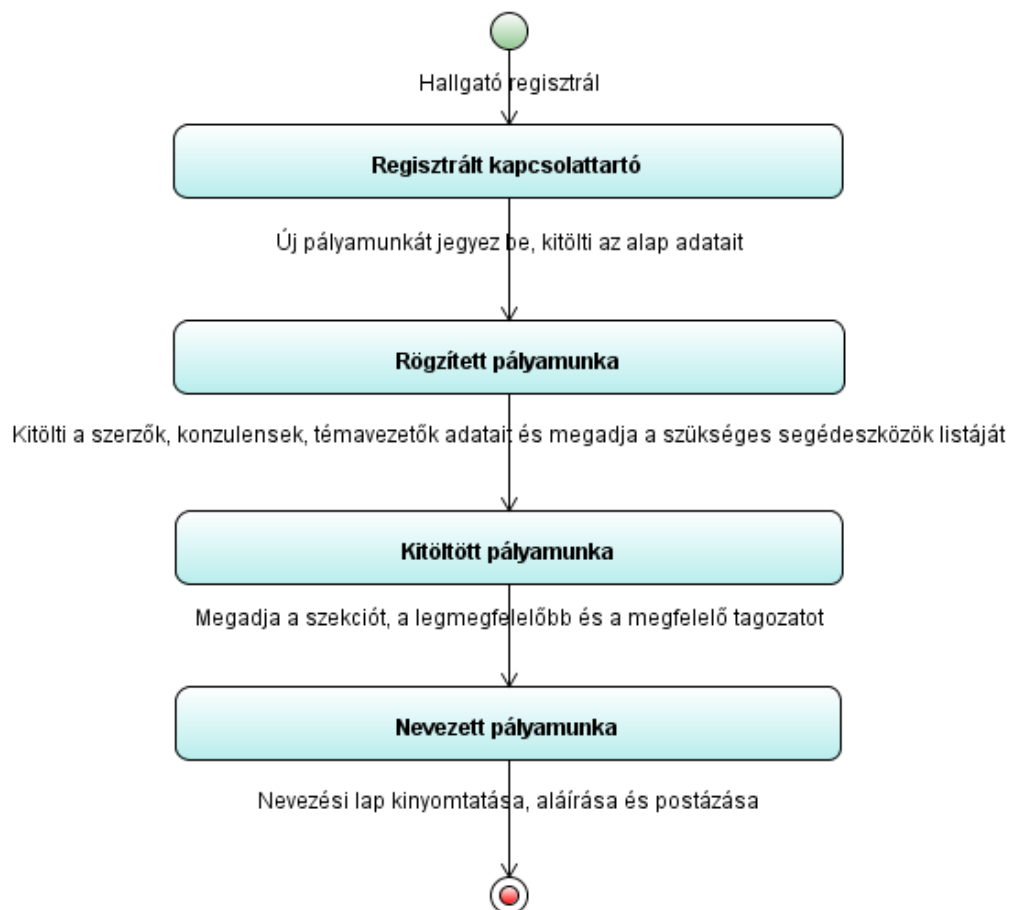
4. ábra: Az intézményi felelősök szerepköreit bemutató UML diagram



5. ábra: A kari felelősök szerepköreit bemutató UML diagram



6. ábra: A hallgatók szerepköreit bemutató UML diagram



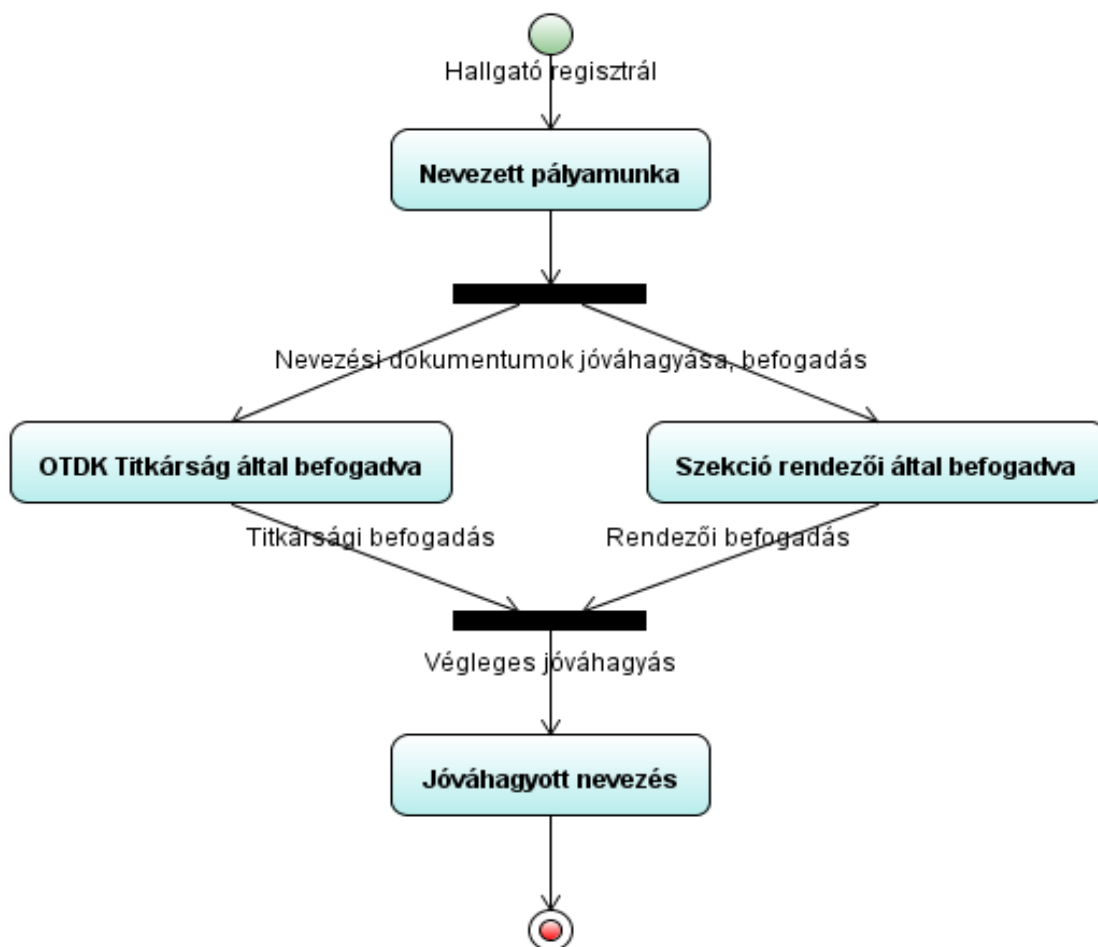
7. ábra: A nevezés folyamatát bemutató UML diagram

2.6 Forgatókönyvek

A legfontosabb tevékenységek:

- **Nevezés:** a nevezés a hallgató feladata. Mivel egy pályamunkának több szerzője is lehet, így gyakorlatban ez a kapcsolattartó dolga. Ő beregisztrál a programba, berögzíti a pályamunka alapvető adatait (cím, intézmény, nyelv,...), és megadja az elkészítésében részt vevő szerzőtársai valamint a konzulensek adatait. Ezek után lehetősége van megadni egy listát arról, hogy a pályamunka bemutatásához milyen technikai eszközökre van szükségük.

Ezek után a program generál neki egy pdf formátumú állományt, a nevezési lapot. Ezt a nevezési lapot ki kell nyomtatnia, és a szerzők aláírásával ellátva el kell juttatnia a szekció rendezőjéhez, valamint a titkársághoz.



8. ábra: A nevezések befogadásának folyamatát bemutató UML diagram

- **Befogadás:** egy nevezés befogadásához több személy elfogadása szükséges. Mind a rendezőknek, mind a titkárságnak vissza kell igazolnia a szükséges nevezési dokumentumok beérkezését, és jelölniük kell az esetleges hiánypótlások szükségességét. Miután minden dokumentum rendben van, jóvá kell hagyniuk a nevezést. Egy nevezés véglegesen akkor tekinthető befogadottnak, ha mind a rendező, mind a titkárság jóváhagyta azt.

2.7 Az adatbázis felépítése

Tekintetbe véve hogy az adatok elérése ORM mapping használatával, LinqToSql osztályok segítségével történik, így az adatbázis nem tartalmaz tárolt eljárásokat, függvényeket, vagy akár triggereket. Ez alól kivételt képeznek a felhasználók kezeléséhez használt, a Microsoft által készített Membership API által generált adatbázis objektumok. Az ORM mapping leegyszerűsítve tulajdonképpen annyit jelent, hogy a relációs adatbázis táblákat megfeleltetjük konkrét objektumoknak, és a táblák kezelését a mapping eszközre, jelen esetünkben a linq-ra bízjuk. Ezáltal az objektumok perzisztensé válnak, és az objektumokon bekövetkezett változások maguk után vonják az adatbázisban történő változtatást is.

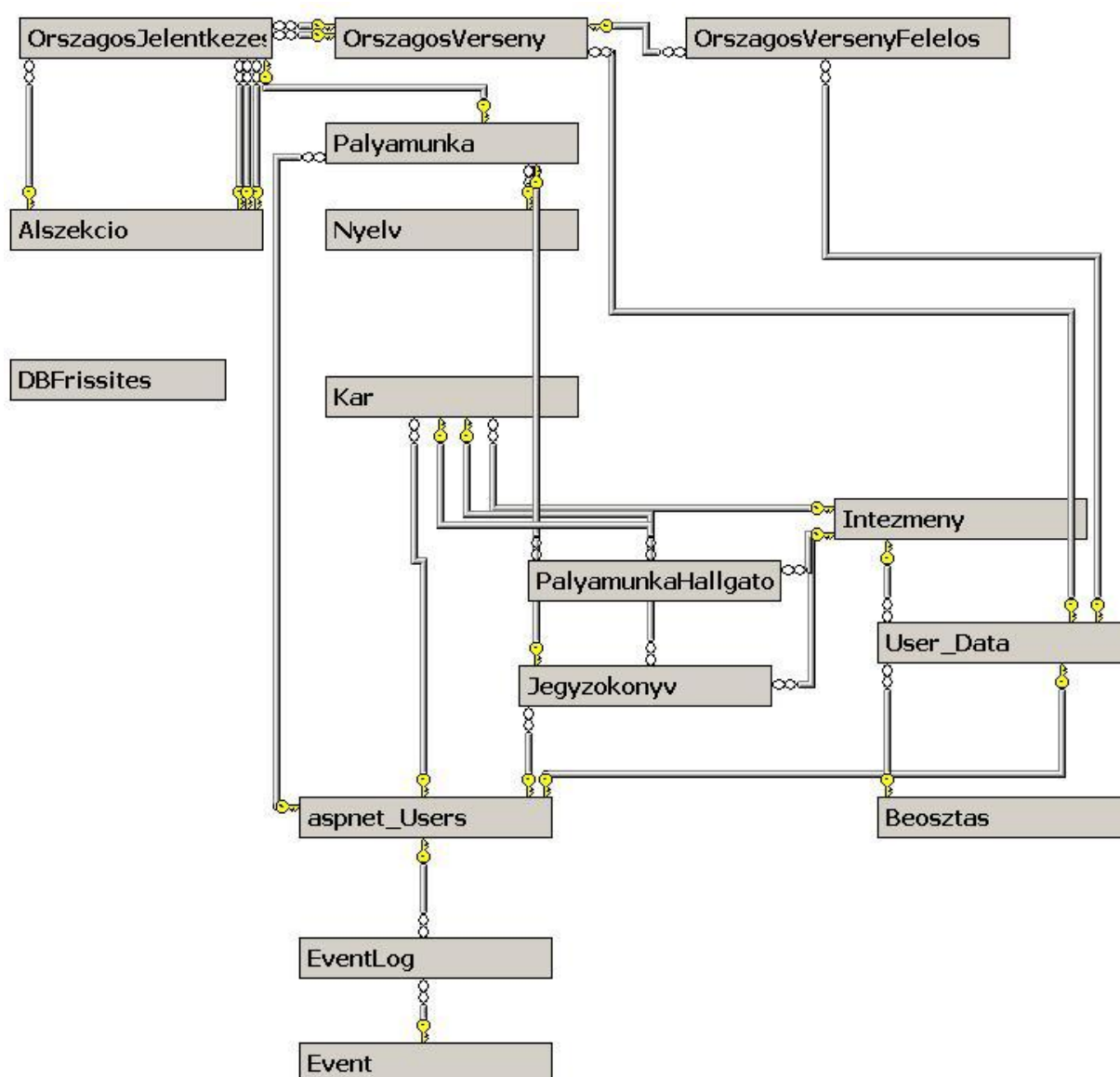
Az adatbázis ilyen módon csak adattárolásra való alkalmazása azért hasznos, mert így élesen elkülönül az üzleti logika az adatoktól. Ezt a határozott elkülönülést akár már egy pár triggerrel is fel lehetne rúgni, és bár tisztában vagyok ezek hatékonyságával, a programkód tiszta és logikus kezelése érdekében elkerültem ezek használatát.

A használt táblák és kapcsolataik: az alábbiakban felsorolom az adatbázisban található relációs táblákat és felvázolom az ezek között kapcsolatokat. A fejlesztés során adatbázis kezeléshez a Microsoft Management Studio Express nevű programot használtuk, így a demonstráláshoz is ebből a programból készített képernyőképeket mutatok.


Elsőként lássuk a táblákat és azok kapcsolatát. A 9. ábrán a szürke téglalapok maguk egy-egy táblát képviselnek. Ezeket szürke vonalak kötik össze egymással, melyek egyik oldalán egy kulcs található. A kulccsal jelölt oldalon található külső kulcs, amely hivatkozik a kapcsolódó tábla azonosító mezőjére.

A táblák részletezése: A táblák ismertetése az egyszerűség kedvéért abc sorrendben történik. Az ábrák megmutatják a táblák oszlopait, valamint ezek típusát. A táblák első sora mindig az elsődleges kulcs, aminek értéke automatikus számozással soronként növekszik.

- **Alszekcio:** Az országos szekciók tagozatainak adatbázisbeli reprezentációja. Az alszekcioID a kulcs, letároljuk még a nevét, valamint hogy aktív-e. Az aktív bit arra szolgál, hogy logikai törlést hajtsunk végre. Ha egy sort passzívvá teszünk, akkor többet nem választható ki a programban, nem rendelhető egyetlen nevezéshez sem.




9. ábra: A táblák kapcsolatait bemutató képernyőkép

Alszekcio			
	Column Name	Data Type	Allow Nulls
	alszekcioID	int	<input type="checkbox"/>
	alszekcioNev	varchar(250)	<input type="checkbox"/>
	szekcioID	int	<input type="checkbox"/>
	aktiv	bit	<input type="checkbox"/>
			<input type="checkbox"/>

10. ábra: Az Alszekcio tábla szerkezetét bemutató képernyőkép


- AspNet_Users:

aspnet_Users			
	Column Name	Data Type	Allow Nulls
	ApplicationId	uniqueidentifier	<input type="checkbox"/>
	UserId	uniqueidentifier	<input type="checkbox"/>
	UserName	nvarchar(256)	<input type="checkbox"/>
	LoweredUserName	nvarchar(256)	<input type="checkbox"/>
	MobileAlias	nvarchar(16)	<input checked="" type="checkbox"/>
	IsAnonymous	bit	<input type="checkbox"/>
	LastActivityDate	datetime	<input type="checkbox"/>
			<input type="checkbox"/>

11. ábra: Az aspNet_Users tábla szerkezetét bemutató képernyőkép

A felhasználók adatainak tárolására használt membership api által generált adatbázistábla. Maga a membership api több táblát is használ, de a program csak ezzel van közvetlen kapcsolatban, így felkerült a listára mivel úgy éreztem, hogy érdemes megmutatni, hogy ez a kapcsolat a generált és a saját kezem által készített táblák között. A membership api tárolja a felhasználók adatait, jelszavait. Ezzel fog kapcsolódni a User_Data tábla, amely a membership api által kínált információkon kívül egyéb, számunkra szükséges adatokat is tárol a felhasználókról.


- **Beosztas:**

Beosztas			
	Column Name	Data Type	Allow Nulls
	beosztasID	int	<input type="checkbox"/>
	beosztasNev	varchar(50)	<input type="checkbox"/>
	aktiv	bit	<input type="checkbox"/>
			<input type="checkbox"/>

12. ábra: A Beosztas tábla szerkezetét bemutató képernyőkép

A felhasználók beosztásának reprezentálására készült. Ilyen jellegű információkra a programban elsősorban a konzulenseknél van szükség.


- **DBFrissites:**

DBFrissites			
	Column Name	Data Type	Allow Nulls
	ID	int	<input type="checkbox"/>
	allomanyNev	varchar(50)	<input type="checkbox"/>
	valtozas	varchar(200)	<input checked="" type="checkbox"/>
	idopont	datetime	<input type="checkbox"/>
			<input type="checkbox"/>

13. ábra: A DBFrissites tábla szerkezetét bemutató képernyőkép

Ez a tábla egy kis technikai segítség a szoftverfejlesztéshez, az alkalmazás futása szempontjából nincs jelentősége. Ebben jegyezzük fel az adatbázis változásaihoz elkészített scripteket és a hatásukról egy rövid összefoglalót. Minden változás sorszámozva van, fejlesztőhöz kötött, megvan a végrehajtásának pontos időpontja. A fejlesztéskor megegyezés volt, hogy az állományneveket készítjük el sorszámozva, és megjelöljük a script készítőjének a monogramjával. Így egyértelműen látszik, hogy időben tekintve milyen sorrendben kell őket lefuttatni, valamint hogy ki a felelős értük. Tehát pl.: az én általam készített 10. script neve: 10_MZs.sql.


- **Event:**

Event			
	Column Name	Data Type	Allow Nulls
	eventID	int	<input type="checkbox"/>
	eventName	varchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

14. ábra: Az Event tábla szerkezetét bemutató képernyőkép

A programban használt logoláshoz szükséges. A különböző bekövetkező események listáját tartalmazza.


- **EventLog:**

EventLog			
	Column Name	Data Type	Allow Nulls
	eventLogID	int	<input type="checkbox"/>
	userID	uniqueidentifier	<input type="checkbox"/>
	eventID	int	<input type="checkbox"/>
	oldValue	varchar(50)	<input checked="" type="checkbox"/>
	newValue	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

15. ábra: Az EventLog tábla szerkezetét bemutató képernyőkép

Ebben a táblában vannak maguk a logbejegyzések. Minden eseményhez tárolja a felhasználó azonosítóját, valamint adatok változása esetén a régi és az új értéket.


- **Intezmeny:**

Intezmeny			
	Column Name	Data Type	Allow Nulls
	intezmenyID	int	<input type="checkbox"/>
	intezmenyNev	varchar(250)	<input type="checkbox"/>
	rovidites	varchar(10)	<input type="checkbox"/>
	varos	varchar(50)	<input checked="" type="checkbox"/>
	iranyitoszam	varchar(6)	<input checked="" type="checkbox"/>
	utca	varchar(100)	<input checked="" type="checkbox"/>
	tdkFelelosID	uniqueidentifier	<input type="checkbox"/>
	aktiv	bit	<input type="checkbox"/>
	virtualis	bit	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

16. ábra: Az Intezmeny tábla szerkezetét bemutató képernyőkép

Az intézmények adatainak reprezentálására szolgáló adatbázistábla. Mindegyik intézménynek van neve, és egy címe. A cím irányítószámra, városra, és utcára bomlik. Az utca mellett tároljuk a házszámot is. A rövidítés megegyezik az OTDK kiadványaiban használt intézményi betűkóddal. Mindegyik intézménynek van egy felelőse, az ő azonosítóját tárolja a felelosID. Az aktív bit természetesen logikai törlésre szolgál. A virtualis egy megjelölő. Speciális esetekben használható. Például lehetséges nevezni országunk határán túlról is. Ezeket a nevezéseket nem intézményeikről közvetlenül fogad be a titkárság, hanem számukra egy virtuális intézményt tart fent, „Határon túli” elnevezéssel.


- **Jegyzokonyv:**

Jegyzokonyv			
	Column Name	Data Type	Allow Nulls
	jegyzokonyvID	int	<input type="checkbox"/>
	intezmenyID	int	<input type="checkbox"/>
	karID	int	<input checked="" type="checkbox"/>
	elnokID	uniqueidentifier	<input type="checkbox"/>
	datumTol	datetime	<input checked="" type="checkbox"/>
	datumIg	datetime	<input checked="" type="checkbox"/>
	zarolt	bit	<input type="checkbox"/>
	iktatoszam	varchar(10)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

17. ábra: A Jegyzokonyv tábla szerkezetét bemutató képernyőkép

Egy helyi verseny jegyzőkönyvét reprezentálja. A jegyzőkönyvek, mint látható intézményeken belül karonként (karID) készülnek el. Mindegyik helyi versenyen volt egy elnök (elnokID). A helyi versenyek tipikusan több napig tartanak, így szükséges az időpontjukat dátum-intervallumhoz kötni (datumTol, datumIg). Amennyiben egy jegyzőkönyv adatai helyesen be vannak rögzítve, le kell zárolni azt (zarolt). Egy adott jegyzőkönyvhöz csak azután köthetőek nevezések, miután zárva lett mivel a nevezéseknél nevezési lapot kell nyomtatni, melyen szerepelnek adatok a helyi jegyzőkönyvekről is. Mindegyik helyi versenynek van egy iktatószáma (iktatoszam) is.


- **Kar:**

Kar			
	Column Name	Data Type	Allow Nulls
	karID	int	<input type="checkbox"/>
	karNev	varchar(250)	<input type="checkbox"/>
	karKod	varchar(10)	<input type="checkbox"/>
	intezmenyID	int	<input type="checkbox"/>
	felelosID	uniqueidentifier	<input checked="" type="checkbox"/>
	aktiv	bit	<input type="checkbox"/>
			<input type="checkbox"/>

18. ábra: A Kar tábla szerkezetét bemutató képernyőkép

Felsőoktatási intézmények karait ábrázolja. Mindegyik karnak van neve (karNev) és egy kódja (karKod). A kód tulajdonképpen egy rövidítés a kar nevére. Mindegyik kar egy intézményhez tartozik (intezmenyID), valamint rendelkezik egy felelőssel (felelosID). Az aktív bit itt is a logikai törlésre szolgál.


- **Nyelv:**

Nyelv			
	Column Name	Data Type	Allow Nulls
	nyelvID	int	<input type="checkbox"/>
	nyelv	varchar(50)	<input type="checkbox"/>
	aktiv	bit	<input type="checkbox"/>
			<input type="checkbox"/>

19. ábra: A Nyelv tábla szerkezetét bemutató képernyőkép

Pályamunka tetszőleges nyelven készíthető. A felhasználható nyelvek listáját tartalmazza. A nyelvet megnevezzük (nyelv). Az aktív bit a logikai törlésre szolgál.

- **OrszagosJelentkezes:**

OrszagosJelentkezes			
	Column Name	Data Type	Allow Nulls
	orszagosJelentkezesID	int	<input type="checkbox"/>
	palyamunkaID	int	<input type="checkbox"/>
	orszagosVersenyID	int	<input type="checkbox"/>
	mejerositve	bit	<input type="checkbox"/>
	legmegfelelobbTagozat	int	<input checked="" type="checkbox"/>
	megfeleloTagozat	int	<input checked="" type="checkbox"/>
	creationDate	datetime	<input type="checkbox"/>
	OTDTiktatoszam	varchar(20)	<input checked="" type="checkbox"/>
	OTDTazonosito	varchar(50)	<input checked="" type="checkbox"/>
	OTDTnevezesiLap	char(1)	<input checked="" type="checkbox"/>
	OTDTrezume	char(1)	<input checked="" type="checkbox"/>
	OTDTtartalmiCimazon...	char(1)	<input checked="" type="checkbox"/>
	OTDTvegzeteknelNyil...	char(1)	<input checked="" type="checkbox"/>
	OTDTosszesitoJegyzek	char(1)	<input checked="" type="checkbox"/>
	OTDThallgatoAlairasa	char(1)	<input checked="" type="checkbox"/>
	OTDTjovahagyas	char(1)	<input checked="" type="checkbox"/>
	OTDTmegjegyzes	varchar(2000)	<input checked="" type="checkbox"/>
	RENDnevezesilapMas...	char(1)	<input checked="" type="checkbox"/>
	RENDrezumeNyomtat...	char(1)	<input checked="" type="checkbox"/>
	RENDrezumeElektroni...	char(1)	<input checked="" type="checkbox"/>
	RENDdolgozat	char(1)	<input checked="" type="checkbox"/>
	RENDjovahagyas	char(1)	<input checked="" type="checkbox"/>
	RENDmegjegyzes	varchar(2000)	<input checked="" type="checkbox"/>
	EredetiOrszagosVerse...	int	<input checked="" type="checkbox"/>
	EredetiLegmegfelelob...	int	<input checked="" type="checkbox"/>
	EredetiMegfeleloTago...	int	<input checked="" type="checkbox"/>
	VanKorhataronTuli	bit	<input type="checkbox"/>
	RENDdolgozatElektron...	char(1)	<input checked="" type="checkbox"/>
	veglegesTagozat	varchar(250)	<input checked="" type="checkbox"/>
	veglegesTagozatID	int	<input checked="" type="checkbox"/>
	helyezes	int	<input checked="" type="checkbox"/>
	kulondij	bit	<input checked="" type="checkbox"/>
	helyezesiMegjegyzes	varchar(250)	<input checked="" type="checkbox"/>
	nemJelentmeg	bit	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

20. ábra: Az OrszagosJelentkezes tábla szerkezetét bemutató képernyőkép

Egy pályamunka jelentkezési információit, a nevezési dokumentumok beérkezésének visszajelzéseit, a jóváhagyásokat, és az országos versenyen elért eredményeket reprezentálja az OrszagosJelentkezes nevű tábla.

Egy adott pályamunkát (palyamunkaID) egy adott országos versenyre (orszagosVersenyID) nevez a hallgató, és megad egy legmegfelelőbb tagozatot (legmegfelelobbTagozatID), és egy megfelelő tagozatot (megfeleloTagozatID). A creationDate mutatja a nevezés időpontját. A nevezés végleges megerősítésekor állítjuk át a megerősítve bit értékét 1-re.

A nevezések befogadásakor a titkárságnak is különböző adatokat kell rögzíteniük. A nevezések kapnak egy iktatószámot (OTDTiktatoszam) és egy OTDT azonosítót (OTDTAzonosito). A titkárságnak meg kell jelölnie, hogy a nevezési dokumentumok beérkeztek-e (i), nem (n), vagy hiánypótlás (h) szükséges. A nevezéshez szükséges dokumentumok: nevezési lap (OTDTnevezesiLap), rezümé (OTDTrezume), a végzett hallgatók esetében nyilatkozat (OTDTvegzetteteknelNyilatkozat) és az intézményi felelősöktől a helyi versenyek összesítő jegyzéke (OTDTosszesitoJegyzek). A titkárságnak be kell jelölnie, hogy a beérkezett dokumentumok tartalmilag megegyeznek-e a címben jelöltekkel (OTDTtartalmiCimazonossag), és a nevezést megjegyzéseikkel láthatják el (OTDTmegjegyzes). A nevezés befogadásánál a titkárság jóváhagyja a nevezést (OTDTjovahagyas).

A rendezőknek is vissza kell jelezniük az elektronikus rendszerben mind a nevezési dokumentumok megérkezését (RENDnevezesiLapMasolat, RENDrezumeNyomtatottan, RENDrezumeElektronikusan, RENDdolgozat, RENDdolgozatElektronikusan), mind a jóváhagyást (RENDjovahagyas), és ők is elláthatják megjegyzésekkel (RENDmegjegyzes) a beérkezett nevezéseket.


Mivel a nevezések elbírálásakor változhat a szekció és a tagozat is, ezért meg kell őrizni a nevezéskor leadott nevezési szekció és tagozat információkat (EredetiOrszagosVersenyID,

EredetiLegmegfelelobbTagozatID, EredetiMegfeleloTagozatID). A nevezéseknél meg kell jelölni, hogy van-e korhatáron túli hallgató a szerzők között (vanKorhataronTuli), mivel ezen jelentkezők külön elbírálásban részesülnek. A korhatáron túliság jelölését a program automatikusan végzi.

A nevezések lezárásakor alakul ki a végleges tagozat (veglegesTagozat, veglegesTagozatID), melyen a dolgozat szerepelni fog az országos megmérettetésen.

Ebben a táblában szerepeltetjük az országos versenyen elért eredményeket is (helyezés, kulondij, helyezesiMegjegyzes), valamint itt kell az országos szekciók rendezőinek megjelölni, ha egy nevezett dolgozat nem került bemutatásra az országos megmérettetésen (nemJelentmeg).

- **OrszagosVerseny:**


OrszagosVerseny			
	Column Name	Data Type	Allow Nulls
	orszagosVersenyID	int	<input type="checkbox"/>
	szekcioID	int	<input type="checkbox"/>
	nevezesiHatarido	datetime	<input type="checkbox"/>
	kezdoDatum	datetime	<input type="checkbox"/>
	vegDatum	datetime	<input type="checkbox"/>
	befejezve	bit	<input type="checkbox"/>
	korhatarMin	int	<input checked="" type="checkbox"/>
	korhatarMax	int	<input checked="" type="checkbox"/>
	felelosID	uniqueidentifier	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

21. ábra: Az OrszagosVerseny tábla szerkezetét bemutató képernyőkép

Az országos versenyeket reprezentáló tábla. Mindegyik verseny egy szekcióban (szekcioID) jelenik meg, és több napos is lehet (kezdoDatum, vegDatum). Amennyiben egy versenyt már lezárt a titkárság, jelölhetjük a befejezve bit használatával. Néhány szekció

korhatárhoz van kötve (korhatarMin, korhatarMax). Mindegyik szekciónak van egy felelőse (felelosID).


- **OrszagosVersenyFelelos:**

OrszagosVersenyFelelos			
	Column Name	Data Type	Allow Nulls
	orszagosVersenyFelel...	int	<input type="checkbox"/>
	orszagosVersenyID	int	<input type="checkbox"/>
	felelosID	uniqueidentifier	<input type="checkbox"/>
			<input type="checkbox"/>

22. ábra: Az OrszagosVersenyFelelos tábla szerkezetét bemutató képernyőkép

A szekciónkénti országos versenyek felelőseit reprezentálja. Megadjuk, hogy melyik szekcióbeli országos versenyhez (orszagosVersenyID) melyik felelős tartozik (felelosID).


- **Palyamunka:**

Palyamunka			
	Column Name	Data Type	Allow Nulls
	palyamunkaID	int	<input type="checkbox"/>
	cim	varchar(500)	<input type="checkbox"/>
	nyelvID	int	<input type="checkbox"/>
	cimIdegenNyelven	varchar(500)	<input checked="" type="checkbox"/>
	kapcsolatTartoID	uniqueidentifier	<input checked="" type="checkbox"/>
	datum	datetime	<input type="checkbox"/>
	javaslat	bit	<input checked="" type="checkbox"/>
	helyezés	int	<input checked="" type="checkbox"/>
	jegyzokonyvID	int	<input type="checkbox"/>
	azonosito	varchar(10)	<input type="checkbox"/>
	zarolt	bit	<input type="checkbox"/>
	megjegyzes	varchar(500)	<input checked="" type="checkbox"/>
	creationDate	datetime	<input type="checkbox"/>
			<input type="checkbox"/>

23. ábra: A Palyamunka tábla szerkezetét bemutató képernyőkép

Egy pályamunka alap adatait tárolja. Az adatok jelentős része a helyi versennyel kapcsolatos, amin a hallgatók megszerezték a jogosultságot az országos megmérettetésen való indulásra. Egy pályamunkának van címe (cím) és jelöljük, hogy milyen nyelven készült (nyelvID). Amennyiben nem magyarul lett megírva, meg kell adni a címét az adott nyelven is (cimIdegenNyelven). Szükséges még a helyi versenyen a pályamunka bemutatásának dátuma (datum), valamint hogy javasolták-e az országos megmérettetésre (javaslat). Meg kell adni a helyi versenyen a helyezését (helyezés), és hogy melyik helyi versenyen került bemutatásra (jegyzokonyvID). A pályamunkáknak van egy azonosítója (azonosito), amely az elektronikus rendszerben használt, és egyedi adat az összes pályaműre nézve. Amennyiben a befogadás során probléma támad, ezzel az információval kereshető elő egy mű a lehető leggyorsabban. A nevezéskor a kapcsolattartó (kapcsolattartoID) megjegyzésekkel (megjegyzes) láthatja el a dolgozatot. Az országos versenyre való megmérettetés előtt zárolnia kell (zarolt), csak ezután adhatja le a nevezését és nyomtathatja a nevezési lapokat. Lejegyezzük azt is, hogy mikor töltötte ki az adatokat az elektronikus rendszerben (creationDate).

- **PalyamunkaHallgato:**


PalyamunkaHallgato			
	Column Name	Data Type	Allow Nulls
	palyamunkaHallgatoID	int	<input type="checkbox"/>
	palyamunkaID	int	<input type="checkbox"/>
	nev	varchar(250)	<input type="checkbox"/>
	szulDatum	datetime	<input checked="" type="checkbox"/>
	szak	varchar(150)	<input checked="" type="checkbox"/>
	szemeszterekSzama	int	<input checked="" type="checkbox"/>
	hanyadikDiploma	char(1)	<input checked="" type="checkbox"/>
	intezmenyID	int	<input checked="" type="checkbox"/>
	karID	int	<input checked="" type="checkbox"/>
	adoszam	varchar(20)	<input checked="" type="checkbox"/>
	szuletesiHely	varchar(50)	<input checked="" type="checkbox"/>
	anyjaNeve	varchar(100)	<input checked="" type="checkbox"/>
	irszam	varchar(6)	<input checked="" type="checkbox"/>
	lakcim	varchar(250)	<input checked="" type="checkbox"/>
	bankszamlaszam	varchar(26)	<input checked="" type="checkbox"/>
	telefonszam	varchar(20)	<input checked="" type="checkbox"/>
	szemeszterekSzamaO...	int	<input checked="" type="checkbox"/>
	abszolutoriumDatum	datetime	<input checked="" type="checkbox"/>
	megjegyzes	varchar(500)	<input checked="" type="checkbox"/>
	mail	varchar(250)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

24. ábra: A PalyamunkaHallgato tábla szerkezetét bemutató képernyőkép

Egy pályamunka szerzőinek adatbázisos reprezentációjára szolgáló adatbázistábla. Mindegyik szerző egy adott pályamunkához köthető (palyamunkaID). Tároljuk a szerzők alapvető adatait: név (nev), születési dátum (szuldatum), születési hely (szuletesiHely), az édesanyja neve (anyjaNeve), lakcím adatok (irszam, lakcim), az eléréséhez szükséges információk (telefonszam, mail); a szerző

intézményéhez köthető információkat: intézmény (intezmenyID), kar (karID); a tanulmányaival kapcsolatosan szükséges adatokat: szak (szak), szemeszterek száma (szemeszterekSzama) és egy megjelölést hogy a hallgató első diplomájához tanul-e (hanyadikDiploma). Vannak olyan adatok, amik az esetleges díjazások gördülékeny intézéséhez szükségesek. Ilyen az adószám (adoszam) és bankszámlaszám (bankszamlaszam). Természetesen a szerzőkhöz is adhatóak megjegyzések.


- **PalyamunkaTechnikaiEszkozok:**

PalyamunkaTechnikaiEszkozok			
	Column Name	Data Type	Allow Nulls
	PalyamunkaTechnikaiE...	int	<input type="checkbox"/>
	PalyamunkaID	int	<input type="checkbox"/>
	TechnikaiEszkozID	int	<input type="checkbox"/>
	megjegyzes	varchar(250)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

25. ábra: A PalyamunkaTechnikaiEszkozok tábla szerkezetét bemutató képernyőkép


Egy pályamunka bemutatásához szükséges technikai eszközök listáját tartalmazó tábla. Meg kell adni, hogy melyik pályamunkához (palyamunkaID) melyik technikai eszközre (technikaiEszkozID) van szükség és itt is tehetünk megjegyzéseket (megjegyzes).

- **Szekcio:**

Szekcio			
	Column Name	Data Type	Allow Nulls
	szekcioID	int	<input type="checkbox"/>
	szekcioNev	varchar(250)	<input type="checkbox"/>
	aktiv	bit	<input type="checkbox"/>
			<input type="checkbox"/>

26. ábra: A Szekcio tábla szerkezetét bemutató képernyőkép


- Az országos versenyek tudományos szekcióinak reprezentációjára szolgáló adatbázistábla. A szekcióknak van nevük (szekcioNev) és logikailag törölhetőek (aktiv).
- **TechnikaiEszkoz:**

TechnikaiEszkoz			
	Column Name	Data Type	Allow Nulls
	technikaiEszkozID	int	<input type="checkbox"/>
	nev	varchar(100)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

27. ábra: A TehcnikaiEszkoz tábla szerkezetét bemutató képernyőkép

A különböző, a dolgozatok bemutatásakor felhasználható technikai segédeszközök listáját tartalmazó tábla. Az eszközök megnevezését a nev oszlop tartalmazza.

- **Temavezeto:**


Temavezeto			
	Column Name	Data Type	Allow Nulls
	temavezetoID	int	<input type="checkbox"/>
	palyamunkaID	int	<input type="checkbox"/>
	nev	varchar(250)	<input type="checkbox"/>
	beosztasID	int	<input checked="" type="checkbox"/>
	intezmenyID	int	<input type="checkbox"/>
	karID	int	<input checked="" type="checkbox"/>
	megjegyzes	varchar(500)	<input checked="" type="checkbox"/>
	tanszek	varchar(250)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

28. ábra: A Temavezeto tábla szerkezetét bemutató képernyőkép

Egy pályamunka létrehozásában segítséget nyújtó témavezetők, konzulensek listáját tartalmazó adatbázistábla. Megadjuk, hogy az adott témavezető melyik pályamunka (palyamunkaID) létrehozásában

segédkezett, mi a neve (nev), mi a beosztása (beosztasID), és hogy melyik intézmény (intezmenyID) melyik karán (karDI), és azon belül melyik tanszéken (tanszek) dolgozik. Természetesen elláthatjuk megjegyzéssel (megjegyzes) is.

- **User_Data:**

User_Data			
	Column Name	Data Type	Allow Nulls
	User_DataId	int	<input type="checkbox"/>
	UserId	uniqueidentifier	<input type="checkbox"/>
	Name	varchar(100)	<input checked="" type="checkbox"/>
	intezmenyID	int	<input checked="" type="checkbox"/>
	beosztasID	int	<input checked="" type="checkbox"/>
	szulDatum	datetime	<input type="checkbox"/>
	anyjaNeve	varchar(100)	<input checked="" type="checkbox"/>
	megjegyzes	nvarchar(500)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

29. ábra: A User_Data tábla szerkezetét bemutató képernyőkép

A felhasználók adatait alapvetően a Membership api által generált táblákban tartjuk, viszont van néhány információ, amire szükségünk van ezeken felül is, így ezeket egy plusz táblában reprezentáltuk, User_Data néven. Mindegyik rekord hozzáköthető egy, a membership api által készített táblához, az aspnet_Users nevűhöz a felhasználók azonosítóján keresztül (UserID). Mindenkinnek van egy neve (nev), és mindenkit intézményhez (intezmenyID) kötünk. Akinek van beosztása (beosztasID), annak azt is letároljuk. Némely esetben az egyértelmű azonosításhoz még szükségünk van a felhasználók születési dátumára (szulDatum), és az édesanyjuk nevére (anyjaNeve) is. Természetesen hozzájuk is tehetünk megjegyzéseket (megjegyzes).

Nézetek: Nézeteket a programban csak néhány helyen használtam, akkor is a megjelenítés megkönnyítése érdekében és elsősorban a fejlesztés kezdeti szakaszaiban, amikor még csak ismerkedtünk az adatok kezelésére használt Linq technológiával.

Később rájöttem, hogy gyakorlatilag nincs szükség rájuk, így egy későbbi nagyobb program átnézésakor ezek törlésre is fognak kerülni amint lehetőség lesz jelentős átvizsgálásra.

A programban jelenleg a következő nézetek vannak:

- vw_FelhasznaloKereso: a felhasználók kereséséhez.
- vw_IntezmenyKereso: intézmények kereséséhez
- vw_JegyzokonyvList: jegyzőkönyvek listája
- vw_PalyamunkaByIntezmeny: Pályamunkák a kapcsolattartó intézményével
- vw_PrintablePalyamunka: Pályamunka nyomtatásához.
- vw_PrintableJegyzokonyv: Jegyzőkönyvek nyomtatásához.
- vw_UserData_Roles: Felhasználók listája.

2.8 Az architektúra

Mint már említettem, az alkalmazás egy többrétegű architektúrára épül. A rétegek éles elkülönítése átláthatóbbá teszi a programot, megkönnyíti a módosításokat, amik a követelmények folyamatos változása miatt igen gyakran előfordulnak a fejlesztés során. Mindegyik réteg kizárólag a közvetlenül alatta állót érheti el, és őt meg csak a közvetlenül felette lévő. A fentebbi rétegek az adatokat közvetlenül az alattuk állóakból nyerik.

Előnye az architektúrának, hogy bármelyik réteg bármikor kicserélhető. Mindössze annyi az elvárás változás esetén, hogy a kicserélt réteg megfelelő metódusokat szolgáltatson. Komplexebb projekteknél célszerűbb lenne ezen követelmény könnyebb megkönnyítése érdekében interfészekkel összekapcsolni a rétegeket, de jelen esetben ezt a projekt mérete miatt nem láttam indokoltnak. Ha például hirtelen fizetőssé válna a linq osztályok használata bármikor kicserélhetnénk az adatelérési réteget mondjuk egy NHibernate (ingyenes ORM eszköz, a javas Hibernate portolása .Net-es alapra) technológiára alapuló megoldássá, akkor az új adatelérési rétegnek akkor biztosítania kellene azt, hogy ugyanazokat az objektumokat szolgáltatassa ugyanazon metódusok segítségével, mint az elődje, és a program hibamentesen üzemelhetne továbbra is.

A programban kivétel bekövetkeztekor a kivétel felgyűrűzik a legfelső, a megjelenítési réteghez, és ott jelenik meg egy addigra már lekezelt hibaüzenet. A kivételek kezelése a bekövetkezésének rétegében történik meg olyan módon, hogy új kivétel keletkezik egy saját üzenettel, és azt dobjuk át a felette található réteg számára.

Megjelenítési réteg

Üzleti logika

Adatelérési réteg

Adatbázis

30. ábra: A program rétegeinek reprezentálása

- **Adatbázis:** Az adatok tárolása itt történik. Az adattároláson kívül különösebb funkciója a programban nincs.
- **Adatelérési réteg:** A programban a DAO (Data Access Object) projektként szerepel. Itt találhatóak az objektum típusok is. Egy DataContext objektum tartalmazza a táblákból generált osztályokat. A DataContext fordítja le a Linq kifejezéseket sql utasításokká, és kéri le az adatokat az adatbázisból. A projekt tartalmaz még statikus osztályokat, melyek csak statikus metódusokat tartalmaznak. Mindegyik táblához, így mindegyik LinqToSql osztályhoz tartozik egy-egy osztály, melybe vannak írva az adatelérési metódusok.
- **Üzleti logika:** A programban BAL (Business Application Layer) projektként szerepel. Ez tartalmazza a program logikai megkötéseit. Akárcsak a DAO, ez is statikus osztályokat tartalmaz.

- **Megjelenítés:** A programban a WebApp (Web Application) projektben található. A nevéből is látszik, hogy a szoftver csak és kizárólag webes felülettel rendelkezik. Komplexebb projekteknél célszerűbb lenne az adminisztrációs feladatokra egy desktop alkalmazás, de itt a szoftver mérete miatt semmi sem indokolja ezt. Mindenesetre, ha egyszer igény lenne rá, megnyugtató érzés tudni, hogy csak egy plusz projektet kellene írni, és nem kellene újra az adatok lekérdezésével foglalkozni egy ilyen változás bekövetkeztekor.

Az eddigieken kívül van a Visual Studio Solutionban néhány további projekt, ezek nem az architektúra szerver részei, hanem a programot segítő dll-eket adnak:

- **AjaxControlToolkit:** ajax-os vezérlők használatára webes oldalon. Ezek lehetővé teszik azt, hogy ne kelljen minden változáskor az egész oldalt betölteni, hanem a webes oldalak részenként változzanak.
- **CustomControls:** Esetleges saját készítésű illetve plusz funkciókkal kiegészített controllokat tartalmaz.
- **DynamicQuery:** Dinamikus Linq kifejezések írásához használható. A Microsoft által készített linq alapvetően csak statikus kifejezések írására képes.

2.9 Konkrét példa (szemléltetésként)

Az eddigi leírások a program konkrét megvalósításával kapcsolatban még nem mondhattak el túl sokat. Nem is ez volt a céljuk, hanem a program nagy léptékben való bemutatása, így a „nagy egésznek” a bemutatása. Ebben a fejezetben nézzük meg a program felépítését közelebbről, a forráskód szintjéről. Egy egyszerűbb adat, a szekció kezelésén keresztül mutatom be, hogy a program elkészítésénél milyen módszereket használtam, hogy hogyan gyűrűznek fel a kivételek és később hogyan kerülnek ezek megjelenítésre. Ezeken túl bemutatásra kerül a példán keresztül az egyes rétegek megvalósítása is. Kicsit belenézünk a már sokat említett Linq lekérdezések használatába és az asp .Net oldalak felépítésébe.

Adatbázis: Mint már említettem, az adatbázisszerverként egy MS SQL 2005-ös adatbázisszervert használunk, így a szekciókat tároló táblákat létrehozó scriptet a szerverhez ingyenesen hozzáférhető SQL Management Studio Express nevű programból másolom ki. Lássuk ebben az egyes szekciók tárolására használt adatbázistáblát a forráskódja szintjén:

```
USE [otdkdb]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

CREATE TABLE [dbo].[Szekcio] (
    [szekcioID] [int] IDENTITY(1,1) NOT NULL,
    [szekcioNev] [varchar](250) NOT NULL,
    [aktiv] [bit] NOT NULL,
    CONSTRAINT [PK_Szekcio] PRIMARY KEY CLUSTERED
(
    [szekcioID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
    CONSTRAINT [UszekcioNev] UNIQUE NONCLUSTERED
(
    [szekcioNev] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO
```

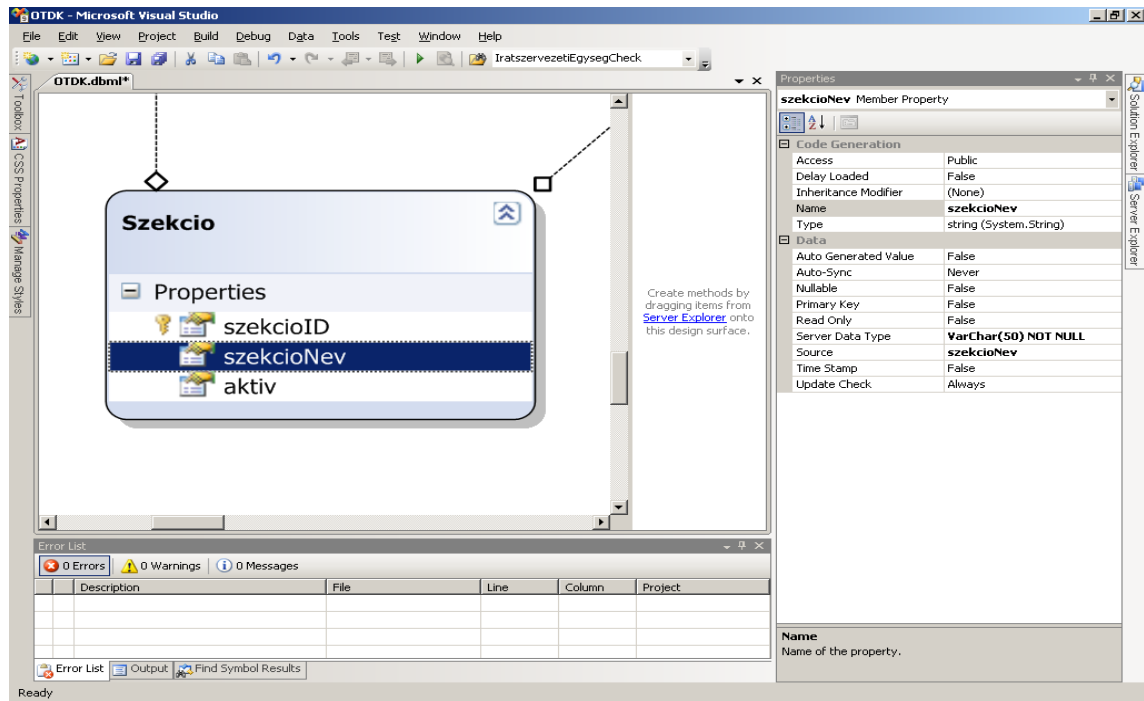
Látható, hogy az elsődleges kulcs egész szám, IDENTITY(1,1) megjelöléssel. Ez a jelzés annyit jelent, hogy a kulcsok sorszámozását az sql szerver végzi az 1-es sorszámtól kiindulva és rekordonként 1-el növeli az azonosító értékét. Ekkor külön utasítás nélkül a szerver nem engedi új sor beszúrását úgy, hogy explicit módon megad hassuk a kulcsot, csak az automatikusan generált sorszámot használja fel. Mivel a tábla minden oszlopára van egy NOT NULL megszorítás, így látható, hogy

mindegyiket kötelező megadni egy szekciót reprezentáló rekordnál. A script egyéb részeire nem kívánok kitérni, mivel ezek már meghaladják ezen dolgozat terjedelmét, természetesen minden egyes rész magyarázata megtalálható a felhasznált irodalmak listáján található SQL Server 2005 Bible című könyvben írásban, valamint az msdn (Microsoft Developer Network) honlapján elektronikus formában.

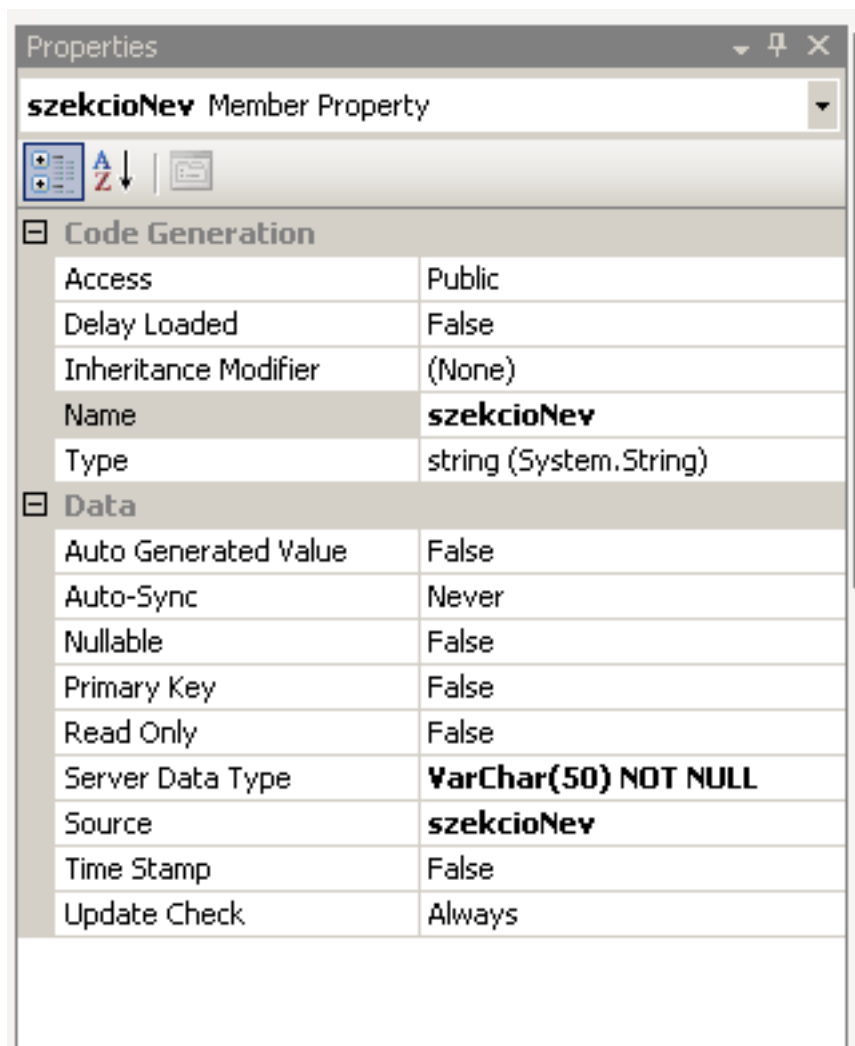
Adatelérési réteg: Most jön a már sokat emlegetett linq bemutatása. A linq önmagában egy lekérdező nyelv, ami a .Net 3.0 framework óta a .Net szerves részévé vált. A Linq egyfajta könnyű, az sql-hez képest átdolgozott logikájú, de mégis az sql-re hasonlító lekérdező nyelv. Gyakorlatilag bármely .Net-ben elérhető nyelv bármilyen kollekció típusú objektumán lehet lekérdezni vele, ami implementálja az IEnumerable interfészt, így egy egyszerű lekérdezéssel lehet akár egy állományrendszerbeli állomány-listát, akár egy adatbázis lekérdezést is kérni. Ennek az eszköznek a kiegészítésére készített a Microsoft egy ORM mapping eszközt, és így születtek meg a LinqToSql osztályok. Ezek segítségével közvetlenül lehet az adatbázisból lekérdezni, és módosító műveleteket végrehajtani az objektumokon perzisztens módon.

A mapping során az adatbázis táblákat konkrét nyelvi osztályoknak feleltetjük meg. Ezen megfeleltetés a Visual Studio 2008-tól kezdve designer támogatással valósul meg, így minimálisra csökkenthető a kézzel begépet programkód sorainak száma. Ennek a bemutatására a már említett fejlesztőeszközben készült képrészleteket mutatok be. Először is lássuk a szekciókat reprezentáló osztályt. Mint látható a 31. ábrán, a szekcio táblához egy Szekcio nevű osztályt hoztunk létre. A tábla minden oszlopát megfeleltettük egy propertynek a Szekcio nevű osztályon belül. Mint a második ezzel kapcsolatos ábra mutatja, az egyes adatbázis oszlopok neve és típusa is le van generálva, így az sql típus megfelelője egy a tárolására alkalmas C# típus lesz. Érdekes a 32. ábráról felhívni a figyelmet a Delay Loaded tulajdonságra. Amennyiben ez true-ra van állítva, akkor a generált lekérdezések az objektumok feltöltésénél nem kérdezik le ezeket az adatbázisból, hanem egyfajta késleltetett betöltés történik. Az adatok akkor kerülnek lekérdezésre, amikor a programban először akarjuk elérni őket. Ez hasznos azokban az esetekben, amikor az adatbázisbeli táblák külső kulccsal kapcsolódnak egymáshoz. Ekkor a mapping szintjén ezek beágyazott objektumokként jelennek meg. Ezek használatakor a beágyazott objektumot tipikusan csak akkor hívjuk le, ha szükség van rá, ez egyszerűen kiváltható a Delay Loaded tulajdonsággal, így nem kell ezekért külön lekérdező utasításokat hívni. Ennek egyetlen megkötése

van, hogy a kapcsolatot nem szabad lezárni. A régebben megszokott programozási nyelvből történő adatkezelésben ökölszabály volt, hogy a lekérdezések végrehajtása után zárjuk le a kapcsolatot. Ez tipikusan úgy történt, hogy a lekérdezést egy try-catch blokkba tettük és a végén a finally blokkban bezártuk a kapcsolatot. Ezen programban a kapcsolat bezárása súlyos hiba lenne, ami hibás adatkezeléshez vezetne.



31. ábra: A szekciókat reprezentáló osztály képe a Visual Studio designeréből



32. ábra: A Szekcio osztály szekcioNev adattagjának tulajdonságai a Visual Studio designeréből

Ezek után tekintsük meg, hogy hogyan történik ténylegesen az adatok elérése. Az adateléréshez használt osztályok csak és kizárólag statikus metódusokat tartalmaznak. A szekciók kezelése a SzekcioDAO osztályban történik. Az osztályban van egy beszűrő, egy módosító, és egy törlő metódus. Mivel több szemszögből is kérdezzük le a programban, így több, hajszálpontosan hat metódus szolgál az adatok lekérésére. Látható a példában, hogy a metódusoknál nyugodtan lezárhatóak a kapcsolatok, ennek egyetlen indoka van: a szekcióknak nincs beágyazott osztályuk, nincsenek benne késleltetett betöltéssel elérhető adatok. A szekciók kezelésénél mindig, minden információ lekérésre kerül az adatbázisszerverről. Maga a programkód:

```
using System;
using System.Collections.Generic;
using System.Linq;
```

```

using System.Text;
using System.Data.Sql;
using System.Configuration;
using System.Data.SqlClient;
using System.Linq.Dynamic;
using System.Data;

namespace DAO
{
    public class SzekcioDAO
    {
        public static List<Szekcio> getSzekcioByOrszagosFelelosID(Guid felelosID)
        {
            OTDKDataContext dc = new OTDKDataContext();
            List<Szekcio> result = null;

            result = (from sz in dc.Szekcio
                      join ov in dc.OrszagosVerseny
                      on sz.szekcioID equals ov.szekcioID
                      where ov.OrszagosVersenyFelelos.Any(ovf => ovf.felelosID ==
felelosID)

                      where ov.befejezve == false
                      select sz).ToList<Szekcio>();
            return result;
        }

        public static List<Szekcio> getAllSzekcio(string sortExp/*, string sortDirection
        */, int startRowIndex, int maximumRows, string searchString)
        {
            OTDKDataContext dc = new OTDKDataContext();
            List<Szekcio> result = null;
            //int allCount = getAllSzekcioCount();
            using (dc)
            {
                result = dc.Szekcio
                    .Where(sz => searchString == null ? 1 == 1 :
sz.szekcioNev.Contains(searchString))
                    .OrderBy(sortExp.Equals("") ? "szekcioNev" : sortExp)
                    .Skip(startRowIndex)
                    .Take(maximumRows).ToList<Szekcio>();
            }

            return result;
        }

        public static int getAllSzekcioCount(string searchString)
        {
            OTDKDataContext dc = new OTDKDataContext();

```

```

        int result = 0;

        using (dc)
        {
            result = dc.Szekcio
                .Where(sz => searchString == null ? 1 == 1 :
                    sz.szekcioNev.Contains(searchString))
                .Count();
        }

        return result;
    }

    public static Szekcio getSzekcioByID(int SzekcioID)
    {
        OTDKDataContext dc = new OTDKDataContext();

        Szekcio szekcio;
        using (dc)
        {
            try
            {
                szekcio = dc.Szekcio.Single(sz => sz.szekcioID == SzekcioID);
            }
            catch (Exception e)
            {
                throw new Exception("Az adott szekció (id: " + SzekcioID.ToString() +
                    ") nem található");
            }
            finally
            {
                dc.Connection.Close();
                dc.Dispose();
            }
        }

        return szekcio;
    }

    public static void deleteSzekcioByID(int SzekcioID)
    {
        OTDKDataContext dc = new OTDKDataContext();

        Szekcio szekcio;
        using (dc) {
            szekcio = dc.Szekcio.Single(sz => sz.szekcioID == SzekcioID);

            dc.Szekcio.DeleteOnSubmit(szekcio);
        }
        dc.SubmitChanges();
    }

```

```

    }

    public static void updateSzekcio(Szekcio Szekcio)
    {
        OTDKDataContext dc = new OTDKDataContext();

        using (dc)
        {
            try
            {
                Szekcio szekcio = dc.Szekcio.Single(sz => sz.szekcioID ==
Szekcio.szekcioID);

                szekcio.szekcioNev = Szekcio.szekcioNev;
                szekcio.aktiv = Szekcio.aktiv;

                dc.SubmitChanges();
            }
            catch (System.Data.SqlClient.SqlException SQLexc)
            {
                switch (SQLexc.Number)
                {
                    case 2627:
                        throw new Exception("A szekciók neveinek egyedinek kell
lenniük!");

                    case 515:
                        throw new Exception("A szekció nevét kötelező kitölteni!");
                    default:
                        throw new Exception("HIBA! " + SQLexc.Number + " " +
SQLexc.Message);
                }
            }
            finally
            {
                dc.Connection.Close();
                dc.Dispose();
            }
        }
    }

    public static void createSzekcio(Szekcio szekcio)
    {
        OTDKDataContext dc = new OTDKDataContext();
        using (dc)
        {
            try
            {
                dc.Szekcio.InsertOnSubmit(szekcio);
                dc.SubmitChanges();
            }
            catch (System.Data.SqlClient.SqlException SQLexc)

```

```

        {
            switch (SQLexc.Number)
            {
                case 2627:
                    throw new Exception("A szekciók neveinek egyedinek kell  
lenniük!");

                case 515:
                    throw new Exception("A szekció nevét kötelező kitölteni!");
                default:
                    throw new Exception("HIBA! " + SQLexc.Number + " " +
SQLexc.Message);
            }
        }
        catch (Exception exc)
        {
            throw new Exception("Egyéb hiba:" + exc.Message);
        }
        finally
        {
            dc.Connection.Close();
            dc.Dispose();
        }
    }
}

public static List<DAO.Szekcio> getSzekcioList()
{
    OTDKDataContext dc = new OTDKDataContext();

    List<Szekcio> result;
    using (dc)
    {
        result = dc.Szekcio
            .Where(sz => sz.aktiv == true)
            .OrderBy(sz => sz.szekcioNev)
            .ToList<DAO.Szekcio>();
    }

    return result;
}

public static List<DAO.Szekcio> getSzekcioListForOrszagos()
{
    OTDKDataContext dc = new OTDKDataContext();
    List<Szekcio> result;
    using (dc)
    {
        var query = from sz in dc.Szekcio
                     join ov in dc.OrszagosVerseny
                     on sz.szekcioID equals ov.szekcioID

```



```

        where sz.aktiv == true && ov.befejezve == false
        //where ov.befejezve == false
        orderby sz.szekcioNev
        select sz;

        result = query.ToList<Szekcio>();
    }
    return result;
}

}
}

```

Látható, hogy a C# nyelvben, ha egyéb névterek eszközeit akarjuk használni, akkor a forrásállomány elején elhelyezett using parancsokkal tehetjük ezt meg. Az is első pillantásra észrevehető, hogy maga az osztály a DAO névtérben található. Maga a DAO névtér pedig a Visual Studio solutionon belül a DAO projektet alkotja. Általános konvenció, hogy a névterek nevei megegyeznek a projektek neveivel, amik tartalmazzák őket.

Érdemes megtekinteni a `getSzekcioListForOrszagos` nevű metódust. Ebben található egy olyan osztály használata, melynek neve `OTDKDataContext`. Ezt tulajdonképpen a Visual Studio generálja a szintén generált `LinqToSql` osztályok kezelésére. Ezen keresztül használhatóak a tényleges objektumok, ennek segítségével fordítja le a linq a megfelelő linq lekérdezéseket adatbázisbeli lekérdezésekkel és a `DataContext` futtatja le az adatbázisbeli queryket és tölti fel a C# objektumokat. A `query` nevű változó értékadásánál látható egy linq lekérdezés. Első pillantásra olyan, mint egy „fejre állított” sql utasítás, mindössze más a sorrendje az egyes részeknek. Ezen kívül még lényeges különbség, hogy ez az sql selecttel ellentétben a linq kifejezés objektumok listáját szolgáltatja. Előre kerül a `from` rész, ahol `join`-okkal megmondjuk, hogy hol keresse az adatokat. Ezután jöhet a `where`, ahol megadjuk, hogy milyen logikai feltételeknek megfelelő objektumok kerülhetnek az eredményhalmazba. Ezt követően jöhet egy `orderby`, vagy egy `orderbyDescending`, ami a sorba rendezést szolgálja. Végül jön a `select`, amivel megadjuk, hogy milyen típusú objektumokat szeretnénk lekérdezni. A `getSzekcioList` nevű metódus mutatja meg a linq lekérdezések egy másik formáját. Itt a legszembetűnőbb a lambda kifejezések használata. Erre most a dolgozat

terjedelme miatt nem kívánok kitérni, mindössze annyit említenék, hogy a lambda kifejezések sokkal inkább átláthatóbbá teszik a lekérdezéseket, mint a hagyományos sql nézet. Maga a .Net framework annyi lehetőséget kínál, hogy a legegyszerűbb, ha mindenki a számára legkézenfekvőbbet használja, én itt most csak két eshetőséget mutattam be, mivel ezek részletezése már elkanyarodna a témától és részletes ismertetése amúgy is terjedelmes lenne egy szakdolgozat kereteihez mérten.

Üzleti logika: A programban az üzleti logika a BAL (Business Application Layer) nevű projektben. A benne található osztályok alkotják a BAL névteret. Akárcsak az adatelérési rétegben, itt is csak statikus metódusokat használunk. Lássuk a szekciók kezelésére használt BAL-beli osztályt:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using DAO;

namespace BAL
{
    public class SzekcioBAL
    {
        public static List<DAO.Szekcio> getAllSzekcio(string sortExpression, /* string
sortDirection, */int startRowIndex, int maximumRows, string searchString)
        {
            return DAO.SzekcioDAO.getAllSzekcio(sortExpression, /*sortDirection,
*/startRowIndex, maximumRows, searchString);
        }

        public static int getAllSzekcioCount(string sortExpression/*, string
sortDirection*/, int startRowIndex, int maximumRows, string searchString)
        {
            return DAO.SzekcioDAO.getAllSzekcioCount(searchString);
        }

        public static Szekcio getSzekcioByID(int SzekcioID)
        {
            return DAO.SzekcioDAO.getSzekcioByID(SzekcioID);
        }

        public static void deleteSzekcioByID(int SzekcioID)
        {
            DAO.SzekcioDAO.deleteSzekcioByID(SzekcioID);
        }
    }
}
```

```

public static List<Szekcio> getSzekcioByOrszagosFelelosID(Guid felelosID)
{
    return DAO.SzekcioDAO.getSzekcioByOrszagosFelelosID(felelosID);
}

public static void updateSzekcio(Szekcio Szekcio)
{
    if (Szekcio.aktiv == false)
    {
        List<Alszekcio> alszekciok =
BAL.AlszekcioBAL.getAlszekcioFullListBySzekcioID(Szekcio.szekcioID);
        foreach (Alszekcio asz in alszekciok)
        {
            asz.aktiv = false;
            BAL.AlszekcioBAL.updateAlszekcio(asz);
        }
        DAO.SzekcioDAO.updateSzekcio(Szekcio);
    }
}

public static void createSzekcio(Szekcio szekcio)
{
    DAO.SzekcioDAO.createSzekcio(szekcio);
}

public static List<DAO.Szekcio> getSzekcioList()
{
    return DAO.SzekcioDAO.getSzekcioList();
}

public static List<DAO.Szekcio> getSzekcioListForOrszagos()
{
    return DAO.SzekcioDAO.getSzekcioListForOrszagos();
}

/*az orszagos versenyre jelentkezett palyamunkahoz
 * adja meg hogy melyik szekcioban versenyzik a palyamunka
 * */
public static DAO.Szekcio getSzekcioForPalyamunka(int palyamunkaID)
{
    return DAO.SzekcioDAO.getSzekcioByID(BAL.OrszagosJelentkezesBAL
.getOrszagosJelentkezesByPalyamunkaID(palyamunkaID).OrszagosVerseny.szekcioID);
}
}
}

```

Az időnként előforduló megjegyzésekből is látszik, hogy maga a fejlesztés most is folyamatban van, egy készülő projektbe csöppentünk bele. Úgy érzem, érdemes kiemelni az updateSzekcio nevű metódust. Ez egy egyszerű példa

arra, hogy az üzleti logikát 100%-ban ebben a rétegben kívántam megvalósítani. Itt tulajdonképpen egy egyszerű követelmény van felvázolva, ami egyébként is természetesen következik már magából az adatok felépítéséből is. Az egyes tagozatok szekciókhoz tartoznak. Ha egy szekciót logikailag törölünk (passzívvá teszünk), akkor a programból nem szabad, hogy elérhetőek legyenek a tagozatai sem, így ezeket is passzív státuszúvá kell állítani. Ezt úgy lehet elérni, hogy egyesével végiglépkedünk a szekció tagozatain és passzívvá tesszük őket. Ez a példa azért is szép, mert látható, hogy ez bizony a hatékonyság rovására megy, hiszen egyesével kell módosítanunk őket. Ezt talán egyetlen darab update utasítás is megtehetné egy adatbázisbeli triggerbe pakolva, viszont akkor ez már megtörné az adatok kezelésének, és a program logikájának éles elkülönülését. Ez egy teljesítmény kritikus alkalmazásnál hiba lenne, de itt már csak a program mérete miatt is elfogadhatónak tekinthető. Ennek az osztálynak további magyarázására nem kívánok kitérni, szerintem az eddigiek alapján a „dallam” már felismerhető, ennek a fejezetnek úgyis a szemléltetés a célja, nem a szoftver pontos leírása, úgyhogy lépünk is tovább a megjelenítési réteghez.

Megjelenítési réteg: Maga az alkalmazás egy több rétegű architektúrára épül. A több rétegnek többek közt az is előnye, hogy a rétegek viszonylag könnyen kicserélhetőek. A megjelenítésnél még jobban kitűnik a több rétegű alkalmazás nagy előnye. Egy programot felvértézhetünk több, egymástól akár technológiailag teljesen elkülönülő megjelenítési felülettel. Készíthetünk egy egyszerű asztali alkalmazást, egy böngészőben futtatható webes programot, de akár egy mobiltelefonra is készíthetünk egy egyszerű wapos felületet, gyakorlatilag csak a különböző platformok véglegessége szab határt a lehetőségek tárházának, és ezen rétegek mind közösen az egyetlen egyszer megírt üzleti logikai rétegből nyerik az adatokat. Jelen program esetében egyelőre csak egy webes felületre volt szükség, az összes felhasználó ezt a felületet használja. Maga a felület egy asp .Net-es webes alkalmazás, ami a WebApp nevű projektben kapott helyet. Nézzük akkor, hogy hogyan néz ki egy webes oldal egy .Net-es webes alkalmazás esetében.

A szekciók kezelésére szolgáló lap, mint minden asp .Net-es oldal gyakorlatilag három állományból áll. A három állomány a szekciók kezelésének esetében:

- SzekcioKarb.aspx
- SzekcioKarb.aspx.cs
- SzekcioKarb.aspx.designer.cs

Ezen állományok közül a harmadik egy, a fejlesztőeszköz által generált állomány, erre most nem térünk ki, így nézzük meg a másik két fület. Az első, a SzekcioKarb.aspx gyakorlatilag a kipakolandó kontrollokat tartalmazza. Ebben adjuk meg az oldal külső megjelenítését. A második fület, a SzekcioKarb.aspx.cs-t nevezi a .Net codebehind-nak. Az elnevezés arra utal, hogy a megjelenítés mögött van egy programkód, ez a codebehind. Ez tartalmazza tulajdonképpen az aspx oldal, és annak kontrolljainak az eseményeit. Tekintsük először magát az aspx-et:

```
<%@ Page Language="C#" MasterPageFile="~/WebPages/Elso.Master" AutoEventWireup="true"
CodeBehind="SzekcioKarb.aspx.cs" Inherits="WebApp.WebPages.Admin.SzekcioKarb"
Title="Szekciók karbantartása" Theme="Primary_skin"%>
<%@ MasterType VirtualPath="~/WebPages/Elso.master" %>
<%@ Register Assembly="CustomControls" Namespace="CustomControls" TagPrefix="ccl" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
<h2 class="myH2">Szekciók karbantartása</h2>

<asp:UpdatePanel ID="UpdatePanel3" runat="server">
<ContentTemplate>
<asp:ObjectDataSource ID="odsSzekcio" runat="server"
SelectMethod="getAllSzekcio"
SelectCountMethod="getAllSzekcioCount"
TypeName="BAL.SzekcioBAL" SortParameterName="sortExpression"
EnablePaging="True">
<SelectParameters>
<asp:Parameter DefaultValue="szekcioNev" Name="sortExpression" Type="String"
/>

<asp:Parameter DefaultValue="0" Name="startRowIndex" Type="Int32" />
<asp:Parameter DefaultValue="10" Name="maximumRows" Type="Int32" />
<asp:Parameter DefaultValue="" Name="searchString" Type="String" />
</SelectParameters>
</asp:ObjectDataSource>
<ccl:gridView ID="gvSzekcio" runat="server" AllowPaging="True"
AllowSorting="True" AutoGenerateColumns="False"
DataSourceID="odsSzekcio" DataKeyNames="szekcioID"
onselectedindexchanged="gvSzekcio_SelectedIndexChanged" Width="510px"
```

```

HorizontalAlign="Center" enableRowSelecting="True" enableRowColoring="True"
defaultSortExpression="szekcioNev" HeaderStyle-HorizontalAlign="Center"
SkinID="gvBaseSkin">
<Columns>
    <asp:BoundField DataField="szekcioID" HeaderText="szekcioID"
        SortExpression="szekcioID" Visible="false" >
    </asp:BoundField>
    <asp:BoundField DataField="szekcioNev" HeaderText="Szekció neve"
        ControlStyle-Width="400px" HeaderStyle-Width="400px"
        SortExpression="szekcioNev" >
    </asp:BoundField>
    <asp:CheckBoxField DataField="aktiv" HeaderText="aktiv" ItemStyle-
HorizontalAlign="Center"
        SortExpression="aktiv" ControlStyle-Width="110px" HeaderStyle-
HorizontalAlign="Center">
    </asp:CheckBoxField>
</Columns>
</cc1:gridView>
</ContentTemplate>
</asp:UpdatePanel>

<div>
<asp:UpdatePanel ID="UpdatePanel2" runat="server">
<ContentTemplate>
    <asp:Table ID="tableForm" runat="server" HorizontalAlign="Center" Width="510px"
Enabled="false">
        <asp:TableRow>
            <asp:TableCell Width="90" HorizontalAlign="Left">
                <asp:Label ID="lblFSzekcioNev" runat="server" Text="Szekció
neve"></asp:Label></asp:TableCell>
                <asp:TableCell>
                    <asp:TextBox ID="tbSzekcioNev" runat="server" Width="400" MaxLength="250">
                    </asp:TextBox>
                    <asp:RequiredFieldValidator ID="rfvSzekcioNev" runat="server"
                        ErrorMessage="Kötelező"
                        SetFocusOnError="true"
                        ControlToValidate="tbSzekcioNev">
                    </asp:RequiredFieldValidator>
                </asp:TableCell>
            </asp:TableRow>
            <asp:TableRow>
                <asp:TableCell>
                    <asp:Label ID="lblFAktiv" runat="server"
Text="Aktív"></asp:Label></asp:TableCell>
                <asp:TableCell>
                    <asp:CheckBox ID="cbAktiv" runat="server" /></asp:TableCell>
                </asp:TableRow>
            </asp:Table>
        </ContentTemplate>
    </asp:UpdatePanel>

```

```

<br />

<asp:UpdatePanel ID="upBSzekcio" runat="server">
    <ContentTemplate>
        <asp:Table ID="tblButtons" runat="server" HorizontalAlign="Center"
Width="510px">
            <asp:TableRow>
                <%--<asp:TableCell HorizontalAlign="Left">
                    <asp:Button ID="bClear" runat="server" Text="Mezők tisztítása"
Width="120" OnClick="clearButtonClick" Enabled="false"/>
                </asp:TableCell--%>
                <asp:TableCell HorizontalAlign="Left" Width="75" >
                    <asp:Button ID="btnInsert" runat="server" Text="Új" Width="70"
OnClick="btnInsert_OnClick"
CausesValidation="false"/>
                </asp:TableCell>
                <asp:TableCell HorizontalAlign="Left" Width="75">
                    <asp:Button ID="btnUpdate" runat="server" Text="Módosítás" Width="70"
OnClick="btnUpdate_OnClick" Enabled="false"
CausesValidation="false"/>
                </asp:TableCell>
                <asp:TableCell HorizontalAlign="Left" Width="75" >
                    <asp:Button ID="btnCancel" runat="server" Text="Mégsem" Width="70"
OnClick="btnCancel_OnClick" Visible="false"
CausesValidation="false"/>
                </asp:TableCell>
                <asp:TableCell HorizontalAlign="Right">
                    <asp:Button ID="btnBack" runat="server" Text="Vissza" Width="70"
PostBackUrl="~/WebPages/index.aspx"
CausesValidation="false"/>
                </asp:TableCell>

            </asp:TableRow>
        </asp:Table>
    </ContentTemplate>
</asp:UpdatePanel>

</div>

</asp:Content>

```

Az oldal elején úgynevezett direktívák vannak. Ezek a direktívák az egész oldalra kihatással vannak. Az elején található egy Page nevű direktíva. Ebben megadjuk, hogy a háttérben található codebehind milyen programozási nyelven készült (`Language="C#"`) magát a működésért felelős codebehind állományt (`CodeBehind="SzekcioKarb.aspx.cs"`), és még több egyéb információt. Egy aspx lap szabvány html (pl. div, table), asp (pl.: asp:GridView, asp:Table) és

saját fejlesztésű (pl.: `ccl: gridView`) kontrollokat is. A `runat=server` attribútumok kezelhetők szerver oldalról, így ezeket érjük el a `codebehind` fileban is. Ez egy szabvány `html` vezérlőre is igaz, bármelyik elérhető szerver oldalról ezen attribútum megadása után. A szerver oldali kontrollok tulajdonképpen a szervernél készülnek el és a klienshez már egy tiszta `html` lap kerül. Ebből következik, hogy mind az `asp` kontrollok, mind a saját készítésű kontrollok alapvető építőkövei a `html` elemek.

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Xml.Linq;

namespace WebApp.WebPages.Admin
{
    public partial class SzekcioKarb : BasePage
    {
        DAO.Szekcio szekcio = null;

        protected void Page_Load(object sender, EventArgs e)
        {
            this.formTitle = "Szekciók karbantartása";

            if (!IsPostBack)
            {
                Log.init();
                Session.Clear();
            }
            else
            {
                {
                    szekcio = Session["selectedSzekcio"] as DAO.Szekcio;
                }
            }
        }

        protected void gvSzekcio_SelectedIndexChanged(object sender, EventArgs e)
        {
            clearControls();
            if (tableForm.Enabled == true)
            {
                tableForm.Enabled = false;
                btnInsert.Text = "Új";
            }
        }
    }
}
```



```

        btnUpdate.Text = "Módosítás";
    }

    try
    {
        szekcio =
BAL.SzekcioBAL.getSzekcioByID(int.Parse(gvSzekcio.SelectedValue.ToString()));

        Session["selectedSzekcio"] = szekcio;

        tbSzekcioNev.Text = szekcio.szekcioNev;
        cbAktiv.Checked = szekcio.aktiv;
        btnUpdate.Enabled = true;
    }
    catch (Exception exc)
    {
        Master.displayErrorMessageOnPopup(exc.Message);
    }
}

protected void btnInsert_OnClick(object sender, EventArgs e)
{
    if (btnInsert.Text == "Új")
    {
        clearControls();
        btnInsert.Text = "Mentés";
        btnInsert.CausesValidation = true;
        btnUpdate.Enabled = false;
        btnCancel.Visible = true;

        cbAktiv.Checked = true;

        gvSzekcio.SelectedIndex = -1;

        tableForm.Enabled = true;
    }
    else
    {
        try
        {
            szekcio = new DAO.Szekcio();
            string szekcioNev = tbSzekcioNev.Text.ToString();
            szekcio.szekcioNev = szekcioNev == "" ? null : szekcioNev;
            szekcio.aktiv = cbAktiv.Checked;
            BAL.SzekcioBAL.createSzekcio(szekcio);
            gvSzekcio.DataBind();
        }
    }
}

```

```

        Log.write(1,
(Membership.GetUser(User.Identity.Name)).ProviderUserKey.ToString(), "---",
szekcio.szekcioNev);

        clearControls();
        gvSzekcio.SelectedIndex = -1;

        Master.displayMessageOnPopup("Új szekció (" + szekcio.szekcioNev + ")
sikeresen felvéve.");

        btnInsert.CausesValidation = false;
    }
    catch (Exception exc)
    {
        Master.displayErrorMessageOnPopup("Egyéb hiba:" + exc.Message);
    }
}

}

protected void btnUpdate_OnClick(object sender, EventArgs e)
{
    if (btnUpdate.Text == "Módosítás")
    {
        btnInsert.Enabled = false;
        btnUpdate.CausesValidation = true;
        btnUpdate.Text = "Mentés";
        btnCancel.Visible = true;

        tableForm.Enabled = true;
    }
    else
    {
        try
        {
            szekcio = Session["selectedSzekcio"] as
DAO.Szekcio;//BAL.SzekcioBAL.getSzekcioByID(int.Parse(gvSzekcio.SelectedValue.ToString()))
;

            string szekcioNev = tbSzekcioNev.Text.ToString();
            szekcio.szekcioNev = szekcioNev == "" ? null : szekcioNev;
            szekcio.aktiv = cbAktiv.Checked;

            BAL.SzekcioBAL.updateSzekcio(szekcio);

            Session["selectedSzekcio"] = szekcio;

            btnInsert.Enabled = true;
            btnUpdate.Text = "Módosítás";
            btnUpdate.CausesValidation = false;
            btnCancel.Visible = false;

```

```

        gvSzekcio.DataBind();

        tableForm.Enabled = false;

        Master.DisplayMessageOnPopup("A szekció (" + szekcio.szekcioNev + ")  
sikeresen módosítva");
    }
    catch (Exception exc)
    {
        Master.DisplayErrorMessageOnPopup(exc.Message);
    }
}

protected void btnCancel_OnClick(Object sender, EventArgs e)
{
    if (btnInsert.Text.Equals("Mentés"))
        clearControls();
    else
    {
        btnInsert.Enabled = true;
        btnInsert.CausesValidation = false;
        btnUpdate.Text = "Módosítás";
        btnCancel.Visible = false;
        tableForm.Enabled = false;
    }
}

protected void clearControls()
{
    btnInsert.Text = "Új";
    btnUpdate.Text = "Módosítás";
    btnInsert.Enabled = true;
    btnUpdate.Enabled = false;
    btnInsert.CausesValidation = false;
    btnUpdate.CausesValidation = false;
    btnCancel.Visible = false;

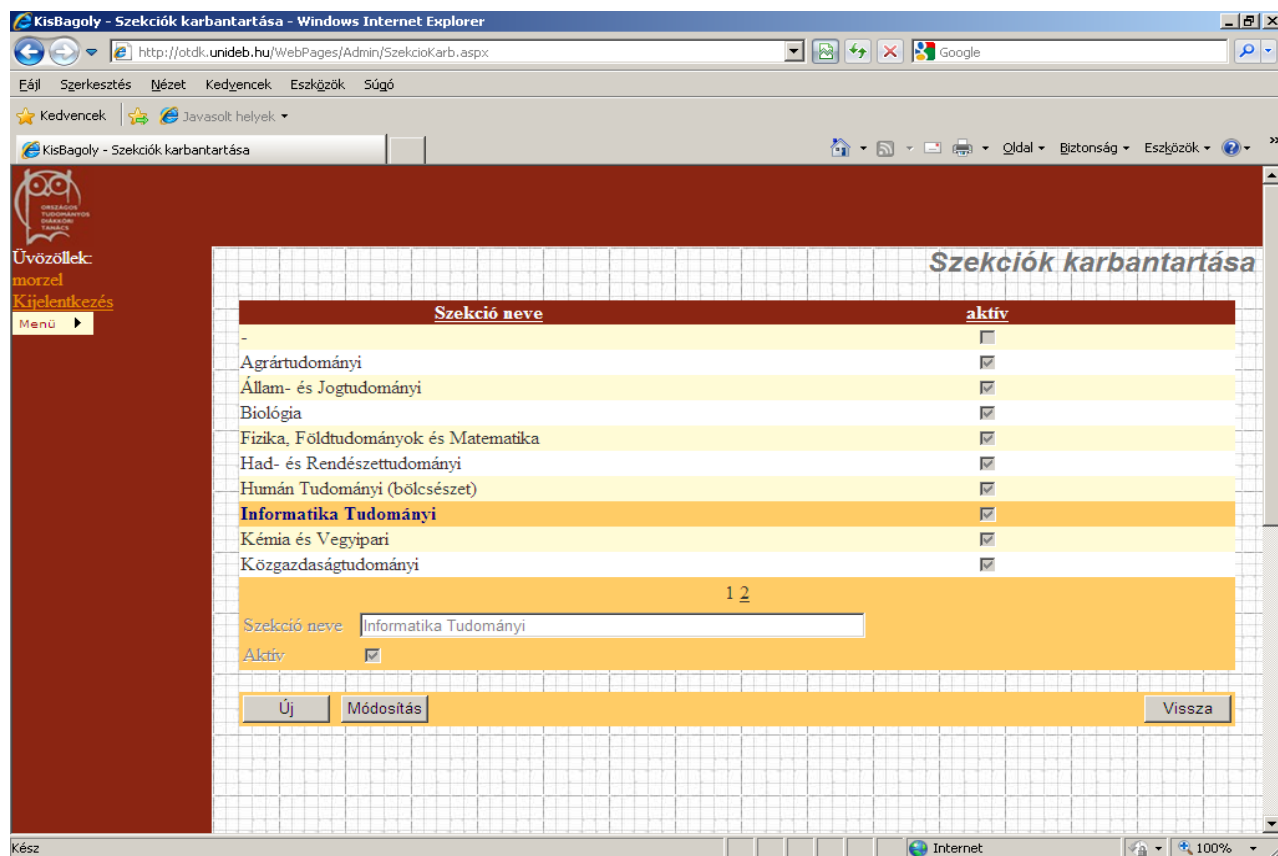
    tbSzekcioNev.Text = "";
    cbAktiv.Checked = false;

    tableForm.Enabled = false;
}
}
}

```

A SzekcioKarb.aspx.cs-ben már egyértelműen látszik, hogy ez tulajdonképpen egy C# nyelvű osztály. Ebben vannak megadva a különböző vezérlők eseményei (pl. gombnyomások hatása), illetve az oldal fontosabb eseményei. Egy aspx

oldalnak úgynevezett életciklusa van, ami leírja, hogy mi történik onnantól kezdve, hogy meg akarjuk jeleníteni odáig, hogy a felhasználó böngészőjében megjelenik egy kész html lap. Ezt nevezzük asp page life cyclennek, magyarul ez az oldal életciklusa. Az életciklus főbb eseményeit bármikor kiegészítheti a programozó további programkóddal, erre példa a `Page_Load`, ami az oldal betöltésekor fut le. Másik fontos eszköz, amire érdemes figyelni, hogy kihasználom a programban még egy előnyös objektumot, melynek neve session. A sessionben látszik az előnye a szerver oldali programozásnak. A session tulajdonképpen egy asszociatív tömb, amelybe objektumokat pakolhatunk. Minden felhasználói munkamenet esetében megszületik egy session, így az adott felhasználóhoz bármilyen adat lementhető ebbe a tömbbe. Ennek első hallásra nincs túl sok haszna, de ne feledjük el, hogy a weben vagyunk, nem desktop környezetben. A weben minden egyes oldal állapota olyan, hogy csak az adott oldalra vonatkozik. Bármilyen esemény bekövetkeztekor az oldal postolásra kerül. A posttal egy időben a szükséges adatok elvesznek. Így ha valamit le akarunk tárolni, akkor azt szoktuk belepakolni a sessionbe, és amikor újra szükségünk van rá, akkor kiolvashatjuk azt. Ez alól kivételt adnak az asp vezérlők. Ezek az adataikat egy másik úton tárolják, és az információk visszaolvasásnál a postback után automatikusan megtörténik. A példánkban is látható, hogy amikor egy konkrét szekcióval dolgozunk, nem akarjuk lekérdezni annak minden adatát minden egyes oldalfrissítés alkalmával, így a szekció kiválasztásánál sessionbe teszem azt, és az oldalfrissítéseknél meg kiolvasom belőle. Természetesen már csak maga egy ilyen aspx oldal működése is megérne egy hosszabb leírást, de sajnos ez sem fér bele a szakdolgozat kereteibe.



33. ábra: A program a szekciók kezelésére szolgáló felületének képernyőképe

3. Az alkalmazás áttekintése

Ezen fejezetben egy rövid áttekintést kívánok nyújtani a programról. A fejezet maga egy leírást nyújt a program menüpontjairól illetve arról, hogy a felhasználók ezeken keresztül milyen funkciókat képesek elérni, mire képesek használni azokat. A függelékben található használati utasítás fog részletesebben kitérni a különböző felületekre, így ott lesznek találhatóak képernyőképek és részletesebb leírások a felületekhez.

Szekciók karbantartása: Az OTDK tudományos szekcióinak a kezelésére szolgáló felület. Itt lehet felvenni új szekciót, itt lehet módosítani a már meglévőket, valamint itt lehet végrehajtani a logikai törlést is. A titkárság tagjai és a rendezők érik el és használhatják. A titkárság bármelyik szekciót kezeli, a rendezők csak azt, amelyiknek a rendezését elvállalták az országos versenyen.

Tagozatok karbantartása: A szekciók tagozatainak kezelésére készült. Ugyanúgy, mint a szekcióknál, itt is csak az alapvető műveletek érhetők el: lehet újat

felvenni, módosítani, illetve logikailag törölni. A titkárság tagjai és a szekciók rendezői érik el. Utóbbiak ugyanazzal a feltétellel, ami a szekciók karbantartása felületnél volt: csak azokat érik el, amelyeknek a rendezésével meg vannak bízva.

Intézmények karbantartása: A programban elérhető intézmények alapvető adatainak kezelésére szolgáló oldal. A titkárság tagjai és az intézményi felelősök használhatják, az intézményi felelősök csak a hozzájuk rendelt sorokat. A titkárság tagjai itt választják ki az intézményi felelősöket. Az alapl műveletek érhetőek el: létre lehet hozni újat, lehet módosítani, és logikailag törölni.

Karok karbantartása: A karok adatainak kezelésére szolgál. A titkárság tagjai, az intézményi felelősök, és a kari felelősök használhatják. A titkárság tagjai bármely intézmény bármely karához hozzáférnek, az intézményi felelősök csak a hozzájuk tartozó intézmény sajátjait, a kari felelősök meg csak a hozzájuk rendelt karokat. A titkárság tagjai valamint az intézményi felelősök itt választhatják ki a kari felelősöket.

Felhasználók karbantartása: A program felhasználóinak a kezelésére szolgáló oldal. Az adminisztrátori jogosultsággal rendelkező felhasználók érik el. Itt lehet új felhasználókat felvenni, módosítani a már meglévő felhasználókat, logikailag törölni őket, és itt lehet jogosultságokat osztani nekik. Az oldal tartalmaz egy lehetőséget, hogy a már meglévő felhasználóknak a program új jelszavat generáljon, és az új jelszót elküldje a felhasználó e-mail címére.

Nevezés az országos versenyre: Ennek a felületnek több funkciója is van. Egyrészt a helyi felelősök itt képesek a helyi versenyek jegyzőkönyvi adatait rögzíteni, másrészt a hallgatók itt tudják egy intézmény, kar és azon belül egy helyi verseny kiválasztása után berögzíteni a pályamunkáiknak az adatait, és ezek után nevezni az egyes műveket.

Jelentkezések állapota: Ez az oldal egy egyszerű adatszolgáltatást nyújt az intézményi felelősök számára, de a titkárság tagjainak is hozzáférhető. Az intézményi felelősök ezen az oldalon tudják ellenőrizni a nevezések befogadásának az állapotát egy webes felületen, illetve karonkénti bontásban pdf formátumban is lekérdezhetik azokat.

Nevezettek listája: A nevezések befogadására illetve az országos versenyen elért eredmények berögzítésére szolgáló oldal. A titkárság tagjai valamint az országos

szekciók rendezői használhatják. A titkárság tagjai és a rendezők itt jelzik a nevezési dokumentumok beérkezését, az esetleges hiánypótlások szükségességét és megjegyzéseikkel láthatják el a nevezéseket. Amennyiben a nevezési dokumentumok rendben vannak, a végleges szekció és tagozat megjelölés után itt hagyhatják jóvá a nevezéseket, és az országos verseny lezajlása után itt lehetséges a verseny eredményeinek berögzítése is. Az adatrögzítésen kívül lehetőség van itt a jelentkezésekről összefoglaló letöltésére pdf formátumban, valamint egy olyan html formában, amiből egyszerűen Excel táblába illeszthetőek az adatok.

Nyomtatási listák: A titkárság számára képes adatokat szolgáltatni a nevezések befogadásának állapotáról. Lehet listázni intézményenként, azon belül karonként, szekciónként, illetve intézményi-szekciónkénti és kari-szekciónkénti bontásban. Az adatokat pdf formátumban szolgáltatja.

Adataim: Minden felhasználó számára elérhető felület. A felhasználók itt képesek módosítani a személyes adataikat és itt tudják módosítani a már meglévő jelszavukat.

Letöltések: A programból esetlegesen elérhető, minden felhasználó számára elérhető állományokat tartalmazza.

4. Összefoglaló és köszönetnyilvánítás, jövőkép

A program fejlesztése egyedülálló lehetőség volt egy átlagos egyetemi hallgató számára. Szemben a képzés erősen elméleti jellegével, volt lehetőségem kipróbálni egy éles alkalmazás fejlesztését és nemcsak követni azt, hanem aktív szerepvállalással részt venni benne a kezdeti ötletektől kezdve az első megbeszéléseken át odáig, amikor már arról kezdhettünk beszélni, hogy mi legyen a program sorsa és feladata az elkövetkező években. Tapasztalatok sokaságával lettünk gazdagabbak, belekóstolhattunk abba, hogy hogyan kell egy program megrendelőjét interjúztatni és kiszedni belőle az információt morzsáinként, abba, hogy milyen az, ha a követelmények egyik napról a másikra 180 fokban fordulnak meg a változásának köszönhetően két este alatt ki kell cserélni a program működését és megérezhettük az elismerés ízét is. A fejlesztés során megismerkedhettünk a Microsoft által nyújtott alapvető technológiákkal, megtanultuk a szerver oldali webes alkalmazások működését és a csapatban történő munkavégzés előnyeit és hátrányait is. Jóleső érzéssel tölt el az, hogy szemben a tanulmányi éveim alatt elkészített sok-sok beadandóval, úgy érzem, hogy valami talán hosszabb ideig megmaradót sikerült létrehozni. Bár maga a program még félkész, de azért már így is elmondhatjuk, hogy jelen szakdolgozat megírásának időpontjában már lezajlott a programunk segítségével a XXIX. Országos Tudományos Diákköri Konferencia a pályamunkák nevezésétől a legtöbb szekció eredményeinek berögzítéséig.

Ezúton szeretném köszönetemet kifejezni dr. Végh János egyetemi tanárnak, aki azzal, hogy összehozott az OTDT titkársággal, nemcsak megteremtette a lehetőséget a program megszületésére, de a fejlesztés során hallgatói státuszomhoz képest hihetetlenül nagy mozgásteret biztosított, így lehetőségem volt személyiségem fejlesztésére is. Szeretném megköszönni még az OTDT titkárság dolgozóinak, köztük különösen Baranyainé Réti Gabriellának, aki képviselte a program megrendelőjét és segített feltámasztani a program követelményeit, képviselte a felhasználók igényeit, de ugyanakkor ki is állt mellettünk, amikor szükség volt rá, és akit az év elején elszólított gyermekének születése, valamint dr. Cziráki Szabinának, aki átvéve Gabi helyét kérdéseim tömkelegére válaszolt, és segít a program jövőjének eltervezésében. Valamint természetesen nem hagyhatom ki a köszöneteim sorából hallgató kollégáimat, Szatmári Andrást és Kocsis Dánielt, akik segítettek a program

elkészítésében, végrehajtották a feladatokat és átvették a projekt vezetését, amíg én elmerülök a szakdolgozat elkészítésében és az államvizsga tételek kidolgozásában.

A program készítése közben fontos tapasztalat volt, hogy egy nap alatt nem lehet megváltani a világot. A készítés során az eredeti, nagyobb volumenű ötletekből igencsak vissza kellett venni, mert rájöttünk, hogy nem lehet az ügyek intézését gyökeresen megváltoztatni az egyik napról a másikra. A szoftver jövője két nagy feladatot jelent számunkra, a fejlesztők számára. Egyrészt mivel az országos verseny lezárultjával kicsit felszabadulnak erőforrásaink, így egy nagy átnézésnek tesszük alá az eddigi rendszert. Végre leellenőrizhetjük minden egyes gomb működését, minden folyamat helyességét, esetleg módosíthatunk az eddigi felületeken a könnyebb felhasználás érdekében. Ezen kívül másik nagy feladatunk eldönteni a folytatás útvonalát. Számomra a legnagyobb eredmény az lenne, ha nem csak az országos versenyre nevezett pályaművek, hanem az előzetesen az egyes intézményekben megrendezett helyi versenyek eredményei is kivétel nélkül szerepelhetnének a rendszerben. Ehhez természetesen meg kell még nyerni a helyi felelősök támogatását, ez önmagában is egy igen jelentős harc, még ha nem is a program soraival, hanem az emberi akarattal. Ezen kívül fontos jövőbeli bővítés lenne az, hogy a hallgatók a pályaműveiket és azok rezüméjét elektronikus formában feltölthessék a szerverünkre és így hosszabb távon az eddigiekhez képest jelentősen lecsökkenthessük a költséges postai forgalmat.

5. Függelék

A program elkészítésében használt eszközök listája:

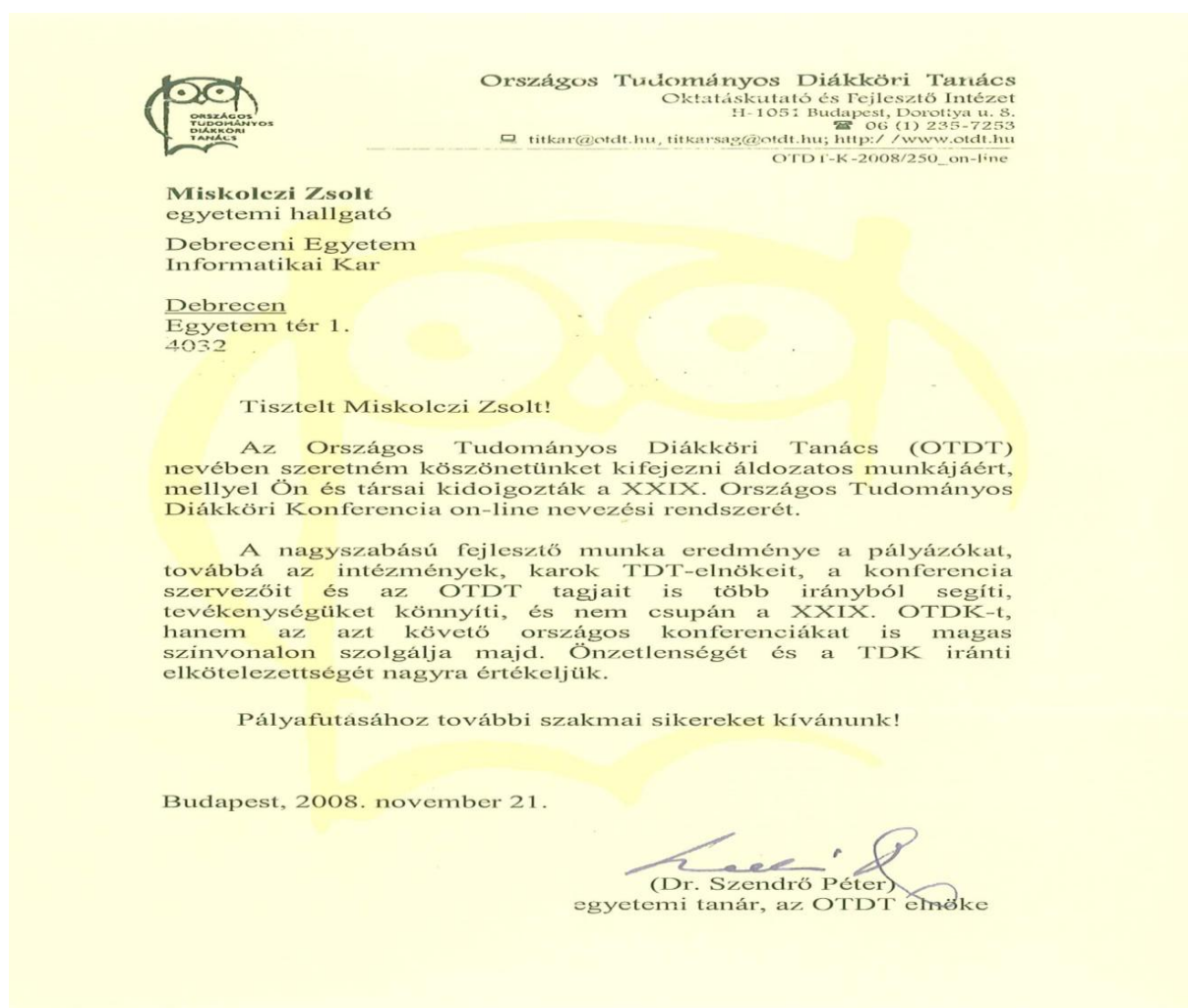
- Microsoft SQL Server 2005
<http://www.microsoft.com/sqlserver/2005/en/us/default.aspx>
- Visual Studio 2008
<http://msdn.microsoft.com/en-us/vstudio/default.aspx>
- iTextSharp
<http://itextsharp.sourceforge.net/>
- Dynamic Linq Libray
<http://weblogs.asp.net/scottgu/archive/2008/01/07/dynamic-linq-part-1-using-the-linq-dynamic-query-library.aspx>

Irodalomjegyzék:

- Paul Nielsen
SQL Server 2005 Bible
- Glenn Johnson and Tony Northup
Microsoft .Net Framework 2.0 Web-Based Client Development Training Kit (MCTS 70-528)
- Stephen Walther
ASP .NET Unleashed
- Reiter István
C# (elektronikus jegyzet, http://people.inf.elte.hu/reiter_i/sharp.html)
- **OTDK anyagok:**
 - Oktatókutató és fejlesztő intézet, OTDT Titkárság
A XXIX. Országos Tudományos Diákköri Konferencia felhívásának dokumentumai
 - Diáktudós – Az Országos Tudományos Diákköri Tanács időszaki kiadványa (2007)

Felhasználói kézikönyv: A felhasználói kézikönyvet terjedelme miatt a szakdolgozat végére tettem.

Elismerések: A programmal kapcsolatban kaptunk néhány olyan szóbeli elismerést, amire büszkék lehetünk. Gondolok itt dr. Pethő Attila nyilvános elismerésére, és nagy büszkeséggel dr. Végh János magánbeszélgetésben kifejezett dicséretére. Dolgozatom zárásaként mellékelem továbbá Dr. Szendrő Péter, az OTDT elnökének köszönő levelének másolatát. Elismerés továbbá, hogy a dr. Pataricza András, az Országos Tudományos Diákköri Tanács Informatikai Szakmai Bizottságának elnöke által felajánlott pezső még mindig gazdára vár. Szendrő Péter, az OTDK elnöke az Informatikai szekció záró ünnepségén köszönte meg a nevezési program elkészítését és üzemeltetését.



34. ábra Az OTDT köszönő levele

FELHASZNÁLÓI KÉZIKÖNYV

AZ OTDK „KISBAGOLY” ON-LINE NEVEZÉSI RENDSZERÉHEZ

TARTALOMJEGYZÉK

Tartalomjegyzék.....	70
Bevezetés	73
A dokumentum felépítése	73
A program telepítése	73
Regisztráció.....	74
TDK felelősök, Szekció rendezők regisztrációja	74
Bírálok regisztrációja	74
Hallgatói regisztráció	74
Bejelentkezés.....	77
Elfelejtett, vagy elveszett jelszó	78
Hallgatók	79
Nevezés az országos versenyre menüpont.....	79
Alap adatok.....	79
Hallgatók.....	81
Konzulensek, témavezetők	82
Technikai eszközök.....	83
Nevezés.....	84
Hibajelentés.....	84
Adataim menüpont	84
Adatok módosítása	85
Jelszó módosítása	85
Pályamunkáim	85
Letöltések menüpont	85
TDK felelősök	86
Az „Adataim” menüpont	86
A „Nevezés az országos versenyre” menüpont	87
Pályamunkák	89
Jelentkezések állapota menüpont	89
Letöltések menüpont	90
Hibajelentés	90
Országos szekció rendezői.....	92
Tagozatok karbantartása.....	92
Új tagozat felvétele.....	92
Tagozat módosítása.....	93
Jelentkezettek listája	93
Általános adatok fül.....	94

Rendezők fül.....	96
Titkárság fül	96
Jelentkezési adatok fül	97
Eredmények fül	97
Adataim	99
Letöltések	99
OTDT titkárság	100
Tagozatok karbantartása.....	100
Intézmények karbantartása	100
Keresés.....	101
Új intézmény felvétele	101
Intézmény módosítása	102
Karok karbantartása	102
Nevezés az országos versenyre.....	103
Jelentkezések állapota	103
Jelentkezettek listája	103
Általános adatok fül.....	104
Rendezők fül.....	105
Titkárság fül	106
Jelentkezési adatok fül	107
Eredmények fül	107
Nyomtatási listák.....	109
Adataim	109
Letöltések	109
Adminisztrátor (admin)	110
Szekciók karbantartása	110
Új szekció felvétele.....	110
Szekció módosítása.....	111
Felhasználók karbantartása	111
Új felhasználó létrehozása.....	111
Felhasználó adatainak módosítása	113
Jelszó újraküldése.....	113

BEVEZETÉS

Jelen felhasználói kézikönyv az OTDK „Kisbagoly” on-line nevezési rendszeréhez készült. A rendszer folyamatos fejlesztés alatt áll, ezért előfordulhatnak hibák és eltérések a kézikönyvben leírtakhoz képest. Amennyiben hibát tapasztal, kérjük, jelezze azt az otdkadmin@gmail.com e-mail címen.

Kérjük, minden esetben írja meg problémájának pontos voltát (lehetőség szerint küldjön képernyőképet is róla).

A DOKUMENTUM FELÉPÍTÉSE

Jelen dokumentum felépítése szorosan kötődik a rendszer felépítéséhez. A kézikönyv főbb fejezetei kapcsolódnak a rendszerben található szerepkörökhöz. A fejezeteken belül az egyes menüpontokat külön tárgyaljuk. Mivel a menüpontok csak néhány funkcióban térnek el a különböző szerepkörű felhasználók szempontjából, ezért előfordulhatnak ismétlődések, vissza- illetve előrehivatkozások.

A rendszerben található szerepkörök az alábbiak:

- Admin
- Hallgató
- Helyi TDT elnök
- Kari TDK felelős
- Országos szekció rendezője
- OTDK bíráló
- OTDT titkárság

A PROGRAM TELEPÍTÉSE

Webes alkalmazásról lévén szó, a program telepítésére nincs szükség.

Használatához szükséges:

- Aktív internetkapcsolat
- Web-böngésző program
- Javascript engedélyezése a böngészőben

Az alkalmazás a <http://otdk.unideb.hu> címen érhető el. Terheltség, hiba esetén használható a <http://otdk.inf.unideb.hu> cím is.

Javasolt/támogatott böngészők:

- Internet Explorer 6.0 vagy újabb
- Mozilla Firefox 2.0 vagy újabb
- Opera 9.0 vagy újabb

REGISZTRÁCIÓ

A <http://otdk.unideb.hu> cím begépelésével a bejelentkező-oldalra jutunk.

Az alkalmazást csak regisztrált, és bejelentkezett felhasználók használhatják.

TDK FELELŐSÖK, SEKCIÓK RENDEZŐINEK REGISZTRÁCIÓJA

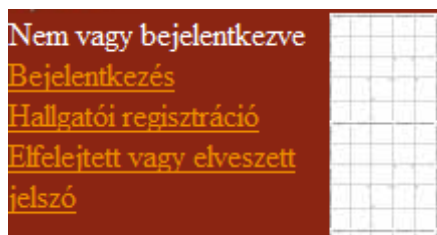
A TDK felelősök, Szekció rendezők adatait az OTDT titkárság rögzíti, illetve regisztrálja őket. Közvetlen regisztrációra nincs lehetőség.

BÍRÁLÓK REGISZTRÁCIÓJA

A bírálói modul jelenleg fejlesztés alatt áll, regisztrációra egyelőre nincs lehetőség.

HALLGATÓI REGISZTRÁCIÓ

Az OTDK-ra regisztrálni kívánó hallgatók ezt a „Hallgatói regisztráció” menüpont alatt tehetik meg.



35. ábra A bejelentkezés nélkül elérhető menüpontok

A „Hallgatói regisztráció” menüpontra kattintva a következő panel jelenik meg:

A screenshot of a student registration form. The form has a yellow background and is set against a grid. It contains six input fields with labels on the left: 'E-mail', 'Teljes név', 'Születési dátum', 'Anyja neve', 'Intézmény', and 'Megjegyzés'. The 'Születési dátum' field has a small calendar icon to its right. The 'Intézmény' field has a dropdown arrow icon to its right. At the bottom right of the form is a button labeled 'Regisztrálok'.

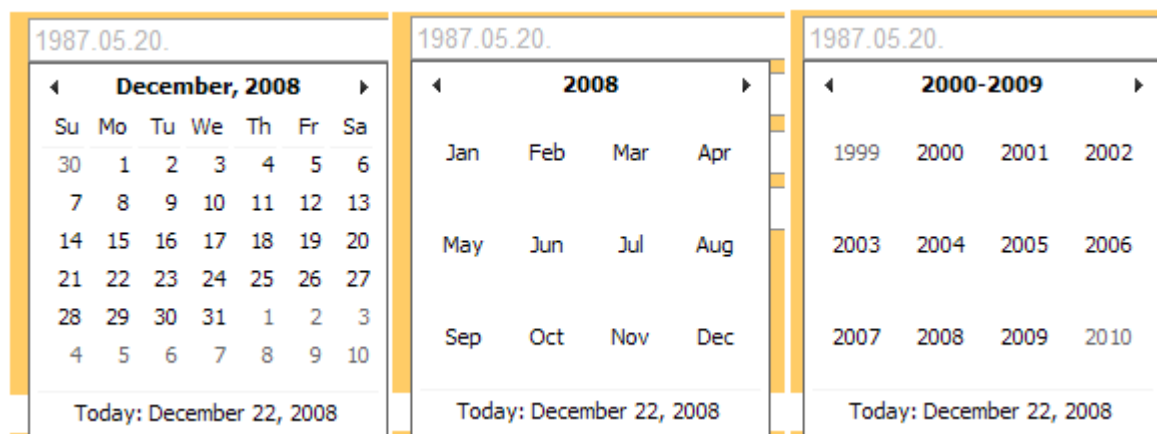
36. ábra. Hallgatói regisztrációs űrlap

Az **E-mail**, **Teljes név**, az **Anyja neve**, illetve a **megjegyzés** mező egyszerű beírással szerkeszthető

Itt megadott e-mail címe fog a későbbiekben felhasználónévként szolgálni, illetve a regisztráció után ide postázzuk a jelszavát.

A **Születési Dátum** megadására a mező melletti **naptár ikonra** kattintva van lehetőség. Ebből kattintással kiválasztható a megfelelő dátum. A gyorsabb navigálás érdekében kattintson a

hónap nevére, ekkor az adott év hónapjaiból választhat. Ha ekkor az évszámmra kattint, akkor évtizedet választhat.



37. ábra. A naptár használata

Az **Intézmény** megadására a mező melletti  gombra kattintva van lehetőség.

Ekkor a megjelenő ablakban az **intézmény nevére kattintva** választhatja az ki. Egy oldalon tíz intézmény jelenik meg. **Lapozásra** a lista alján található számokra kattintva van lehetőség.

Választását a **Kiválasztás** gombra kattintva véglegesítheti, ekkor az intézmény neve megjelenik az előző lapon.

Intézmény	Felelős	Rövidítés	Város	aktív
Intézményen kívüli	Szatmári András	-	-	<input checked="" type="checkbox"/>
A Tan Kapuja Buddhista Főiskola	Farkas Attila Márton	TKBF	Budapest	<input checked="" type="checkbox"/>
Általános Vállalkozási Főiskola	Dr. Szalai Ibolya	ÁVF	Budapest	<input checked="" type="checkbox"/>
Apor Vilmos Katolikus Főiskola	Kádár Gyula	AVKF	Vác	<input checked="" type="checkbox"/>
Budapesti Corvinus Egyetem	Dr. Pénzes Béla	BCE	Budapest	<input checked="" type="checkbox"/>
Budapesti Gazdasági Főiskola	Dr. Harmat Zsigmond	BGF	Budapest	<input checked="" type="checkbox"/>
Budapesti Kommunikációs és Üzleti Főiskola	Dr. Róka Jolán	BKF	Budapest	<input checked="" type="checkbox"/>
Budapesti Műszaki és Gazdaságtudományi Egyetem	Dr. Kovács Ádám	BME	Budapest	<input checked="" type="checkbox"/>
Budapesti Műszaki Főiskola	Dr. Borbély Endre	BMF	Budapest	<input checked="" type="checkbox"/>
Debreceni Egyetem		DE	Debrecen	<input checked="" type="checkbox"/>

1 2 3 4 5 6

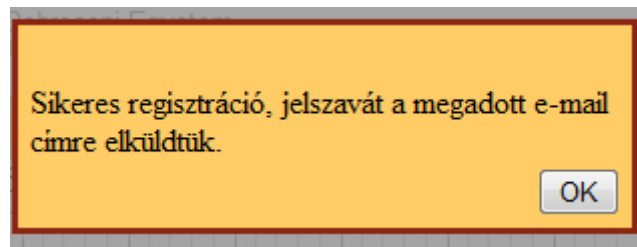
Keresés

Kiválasztás

38. ábra. Intézménykereső

A megjegyzés mezőn kívül minden kitöltése kötelező.

Regisztrációját a „Regisztrálok” gomb megnyomásával véglegesítheti. Sikeres regisztráció esetén a következő üzenetet kapja:



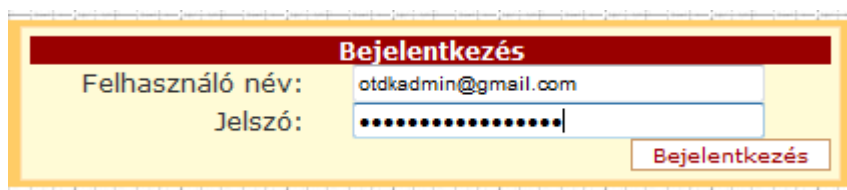
39. ábra. Üzenet a sikeres regisztrációról

Az üzenetet az OK gombra kattintva zárhatja be.

Amennyiben ez az üzenet nem jelenik meg, kérjük, ellenőrizze adatait.

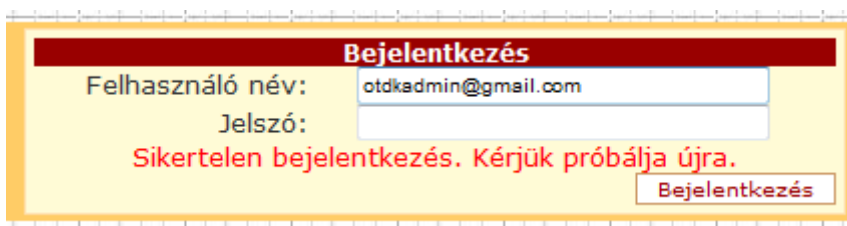
BEJELENTKEZÉS

A bejelentkező oldalon e-mail címét és jelszavát megadva jelentkezhet be.



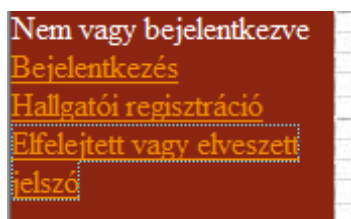
40. ábra. Bejelentkezés

Az esetleges sikertelen bejelentkezésről tájékoztatást kap:



41. ábra. Sikertelen bejelentkezés

Többszöri sikertelen bejelentkezés esetén elképzelhető, hogy rossz jelszót adott meg. Ekkor kattintson az „Elfelejtett vagy elveszett jelszó” hivatkozásra.



42. ábra. Elfelejtett Jelszó

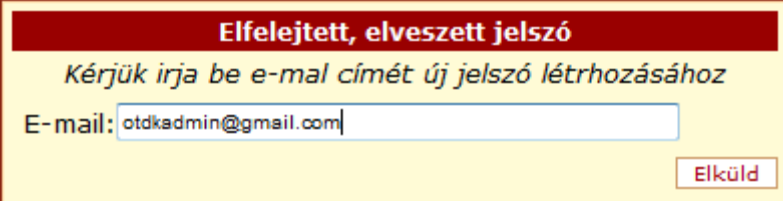
Kérjük, többszöri sikertelen bejelentkezés esetén ne próbálkozzon újra, mivel **a tizedik sikertelen bejelentkezés után a rendszer biztonsági okokból kitiltja.**

Amennyiben többszöri próbálkozás után, és a jelszó visszaszerzése után sem sikerül belépnie a rendszerbe, kérjük, ezt jelezze a titkar@otdt.hu vagy az otdkadmin@gmail.com címen.

Bejelentkezés után a nyitóoldalra kerül, ahol a friss híreket, információkat olvashat, illetve elérheti a menüt. A menü összetétele szerepkörönként változik, erről részletesen a későbbi fejezetekben.

ELFELEJTETT, VAGY ELVESZETT JELSZÓ

E-mail címe megadásával lehetősége van új jelszót kérni.



Elfelejtett, elveszett jelszó

Kérjük írja be e-mail címét új jelszó létrehozásához

E-mail:

Elküld

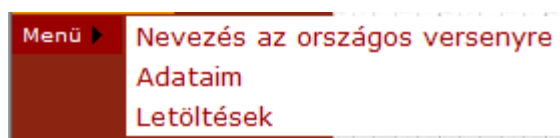
43. ábra. Új jelszó kérése

Új jelszavát a megadott e-mail címre postázzuk.

HALLGATÓK

Bejelentkezés után a következő menüpontok érhetők el:

- Nevezés az országos versenyre
- Adataim
- Letöltések



44. ábra. A hallgatók által elérhető menüpontok

NEVEZÉS AZ ORSZÁGOS VERSENYRE MENÜPONT

Ezen az oldalon nevezheti pályamunkáját a kívánt szekcióba az alábbiakban leírt módon.

Pályamunka regisztrálásához első lépésként ki kell választania azt az intézményi/kari TDK konferenciát, amelyen munkáját bemutatta. Ezt az alábbi lépéseket követve teheti meg:

Válassza ki azt az intézményt a legördülő listából, ahol részt vett TDK konferencián! (Középiskolás tanulók aszerint, hogy mely konferencián szerezték a jogosultságot a XXIX. OTDK-ra történő nevezésre az 'Országos Középiskolai Diákköri Konferenciát' vagy a „Kutató Diákokért Alapítványt” válasszák ki és NE azt az intézményt, ahol a konferencia zajlott, a megjegyzés rovatban pedig írják meg pontosan középiskolájuk nevét.)

Amennyiben az intézményhez tartoznak karok, a legördülő listából válassza azt, ahol részt vett a TDK konferencián. Egyébként a „Nincs kar” feliratú kart válassza.

Az intézmény, illetve kar kiválasztása után külön sorban megjelennek az ott rendezett TDK konferenciák időpontjai (kezdő- és vég dátum), valamint a TDK elnök neve. Az adatsor egy-egy ún. TDK jegyzőkönyvet takar. Ezek közül a dátum alapján ki kell választania, hogy melyiken vett részt.

Választása után a kék színnel megjelenő **Pályamunkák** felíratra kattintva veheti fel saját munkájának adatait, illetve módosíthatja a már előzőekben rögzített műveit.

Intézmény:	Debreceni Egyetem	
Kar:	Informatikai Kar	
Kezdődátum	Végdátum	Elnök
2007:12:06	2007:12:06	Dr. Végh János
2008:11:27	2008:11:27	Dr. Végh János

[Pályamunkák](#)

[Pályamunkák](#)

45. ábra. Jegyzőkönyv kiválasztása

ALAP ADATOK

Pályamunka rögzítéséhez kattintson az **Alap adatok** fül alján található **Új** gombra. Több szerzős pályamunka esetén, kérjük, hogy egy pályamunka csak egyszer kerüljön rögzítésre a rendszerben! A hallgatók adatsorán belül minden szerző adatának rögzítésére van lehetőség.

Tudnivalók az **Alap adatok** megadásához:

- **Azonosító:** csak betűket (ékezetek nélkül) és szóközöket tartalmazó, max. 10 karakterből álló, egyénileg kitalált azonosító.
- **Cím:** kérjük a pályamunka címének pontos megadását (egyeznie kell a rezümé címével).
- **Nyelv:** legördülő listából kiválaszthatja a pályamunka nyelvét.
- **Cím idegen nyelven:** kérjük, amennyiben nem magyar nyelven készült pályamunkája, adja meg idegen nyelvű címét.
- **Jogosultság megszerzésének időpontja:** kérjük, hogy azon intézményi vagy kari konferencia pontos időpontját adja meg, amelyen szerepelt. Ezt a mező mellett található naptár ikonra kattintva teheti meg (az évek gyors váltásában segítségére van, ha az évszámra kattint). A dátum megadása a mezőbe történő beírással nem lehetséges.
- **Javaslat:** a négyzetben elhelyezett jelzéssel erősítheti meg, hogy az intézményi, kari zsűri javasolta az adott pályamunka részvételét a XXIX. OTDK-ra.
- **Helyezés:** megadhatja az intézményi, kari konferencián elért helyezését (nem minden esetben adnak ki helyi konferencián helyezést, így nem kötelezően kitöltendő adat).
- **Megjegyzés:** kérjük, itt jelezze, ha valamilyen adatának kitöltése sikertelen, vagy a kitöltött adatsorhoz egyéb hozzáfűznivalója van.

Amennyiben már előzőekben rögzített pályamunkájának Alap adatait szeretné módosítani, válassza ki azt az oldal tetején található listából a megfelelő címet és kattintson a **Módosítás** gombra.

Az adatok megadása után kattintson a **Mentés** gombra, ekkor pályamunkája megjelenik az oldal tetején található listában.

Amennyiben nem kap külön „ablakban” visszajelzést arról, hogy a Pályamunka mentése megtörtént, illetve a **Mentés** gomb felirata nem változik meg, úgy kérjük, ellenőrizze a megadott adatokat. A kötelezően kitöltendő mezőkre piros csillag (*) figyelmeztet, az egyéb hibákra szöveges üzenetek.

A **Törlés** gomb segítségével törölheti pályamunkáját, amíg még nem jelentkezette azt országos versenyre. A jelentkezés után módosítás nem lehetséges, ezért kérjük többször is ellenőrizze a megadott adatokat.

Miután Pályamunkáját elmentette, a **Hallgatók**, a **Konzulensek**, **témavezetők**, és a **Technikai eszközök** füleken adhatja meg a pályamunka további adatait.

FIGYELEM!!! Az **Új** és **Módosítás** gombok működése ezeken a füleken megváltozik.

46. ábra. Az "Alap adatok" fül

HALLGATÓK

A **Hallgatók** fülön rögzítheti a Pályamunka készítőinek adatait. Az elmentett hallgató(k) adatai a fülek alatt lévő listában jelennek meg. A gombok működése hasonlóan, a fentebb már említett módon figyelmeztet az adatok hibáira, illetve hiányosságaira.

Tudnivalók a **Hallgatók** adatainak megadásához:

- **Név, szak:** kötelezően kitöltendő adatok.
- **Az országos nevezés idején aktuális aktív félév száma** és a **Végzett hallgatók esetén az abszolutórium pontos dátuma:** csak az egyik adat kitöltése kötelező. Amennyiben még aktív státusza van az intézményben, kérjük a félév számát adja meg, ha pedig már végzett hallgató, írja be az abszolutórium pontos dátumát.
- **Diploma:** válasszon a legördülő menüből, hogy első vagy második diplomás.
- **Születési dátum, születési hely, anyja neve, irányítószám, cím, e-mail:** az egyén azonosításához és a kapcsolattartáshoz szükséges kötelező adatok.
- **Telefonszám, bankszámlaszám, adószám:** nem kötelező adatok, a konferencia során az esetleges pénzdíjak kifizetését könnyítik meg.
- **Megjegyzés:** kérjük, itt jelezze, ha valamilyen adatának kitöltése sikertelen, vagy a kitöltött adatsorhoz egyéb hozzáfűznivalója van

FONTOS!!! Mielőtt hallgatót vinne fel, győződjön meg arról, hogy pályamunkája ki van választva az oldal tetején lévő listából. Amennyiben nincs, válassza ki, csak ezután tud hallgatót felvenni. (A kiválasztáshoz kattintson a pályamunka címére. A kiválasztott elem a többitől vastagabb kék betűkkel van szedve, és narancs a háttere).

Cím	Nyelv	Javaslat	Helyezés
teszt	magyar	<input checked="" type="checkbox"/>	1

Alap adatok
Hallgatók
Konzulensek, témavezetők
Technikai eszközök

Név	Születési dátum
test	1987.05.20

Név:

test

Szak:

PTM

Az országos nevezés idején aktuális aktív félévek száma:

3

Végzett hallgatók esetén az abszolutórium pontos dátuma:

Diploma:

első ▾

Születési dátum:

1987.05.20.

Születési hely:

test

Anyja neve:

test etel

Irányítószám:

4000

Cím (Város, utca, házszám):

test

Telefonszám:

E-mail:

test@1.hu

Bankszámlaszám:

Adószám:

Megjegyzés:

47. ábra. A "Hallgatók" fül

KONZULENSEK, TÉMAVEZETŐK

A **Konzulensek, témavezetők** fülön a Hallgatókhoz hasonlóan rögzítheti a Pályamunkájának konzulenseit, témavezetőit.

Tudnivalók a **Konzulensek, témavezetők** adatainak megadásához:

- **Név:** amennyiben konzulense „Dr.”, kérjük a neve előtt feltüntetni azt.
- **Beosztás:** kérjük, a legördülő menüből válassza ki a megfelelő beosztást. Amennyiben az egyéb kategóriát választja, a megjegyzés rovatba beírhatja pontos megnevezést.
- **Intézmény, kar:** az előző adatsorokban megismert módon tölthető ki.
- **Tanszék:** kérjük a pontos megnevezést.
- **Megjegyzés:** kérjük, itt jelezze, ha valamilyen adatának kitöltése sikertelen, vagy a kitöltött adatsorhoz egyéb hozzáfűznivalója van.

FONTOS!!! Hasonlóan a hallgatókhoz itt is ellenőrizze, hogy a pályamunka ki van-e választva.

Cím	Nyelv	Javaslat	Helyezés
teszt	magyar	<input checked="" type="checkbox"/>	1

Alap adatok | Hallgatók | Konzulensek, témavezetők | **Technikai eszközök**

Név	Beosztás
teszt	foiskolai tanár

Név:
Beosztás:
Intézmény:
Kar:
Tanszék:
Megjegyzés:

48. ábra. A "Konzulensek, témavezetők" fül

TECHNIKAI ESZKÖZÖK

A **Technikai eszközök** fülön a legördíthető kínálatból választhatja ki a pályamunka bemutatásához igényelt technikai eszközöket.

Cím	Nyelv	Javaslat	Helyezés
teszt	magyar	<input checked="" type="checkbox"/>	1

Alap adatok | Hallgatók | Konzulensek, témavezetők | **Technikai eszközök**

Eszköz
projektor

Eszköz:
Megjegyzés:

49. ábra. A "Technikai eszközök" fül

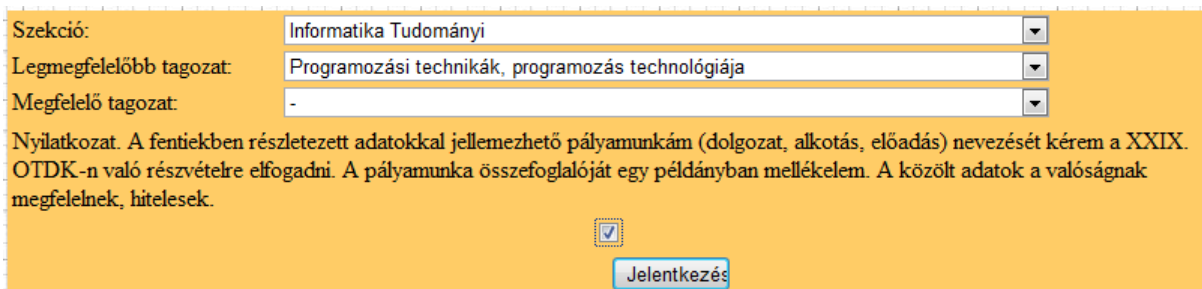
Megjegyzés: kérjük, itt jelezze, ha valamilyen adatának kitöltése sikertelen, vagy a kitöltött adatsorhoz egyéb hozzáfűznivalója van.

FONTOS!!! A pályamunka adatainak rögzítése nem egyenértékű a nevezéssel. A nevezést az alap adatok fülön teheti meg a panel alján megjelenő **„Jelentkezés az országos versenyre”** felíratra kattintva. **Részletek lentebb.**

FIGYELEM: mielőtt a **„Jelentkezés az országos versenyre”** gombot megnyomja, ellenőrizze az adatait, mert a jelentkezés után már **NEM** tud módosítani, illetve nem törölheti a jelentkeztetett pályamunkát.

NEVEZÉS

Miután pályamunkája valamennyi kötelezően kért adatát megadta, térjen vissza az **Alap adatok** fülre, amelynek az alján találja a **Jelentkezés az országos versenyre** gombot. Erre kattintva, a megjelenő ablakban jelentkezetheti Pályamunkáját az országos versenyre. Itt kiválaszthatja a szekciót, azon belül pedig a leginkább megfelelő és a megfelelő tagozatokat. A sikeres jelentkezésről a program tájékoztat. A bezár gombbal visszatérhet a pályamunka adataihoz.



50. ábra. Pályamunka nevezése

Ezt követően szintén az Alap adatok fül alján található **Jelentkezési lap nyomtatása** gombra kattintva kinyomtathatja a jelentkezési lapot (ez a Jelentkezés az országos versenyre gomb helyén jelenik meg). A gomb megjelenéséhez szükség lehet a böngésző frissítésére (F5 billentyű megnyomásával elérhető, a CTRL és F5 billentyűk lenyomásával frissítheti a teljes oldalt. Esetleg újra szükség lehet a pályamunka kiválasztására is.).

HIBAJELENTÉS

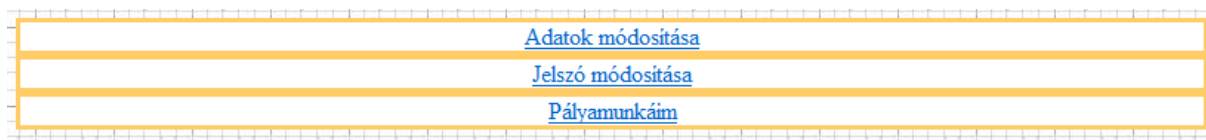
A rendszer működésével kapcsolatos kérdéseivel, észrevételeivel forduljon a fejlesztőkhöz az otdkadmin@gmail.com címen. Hiányzó időpontok, intézmények, konferenciák és egyéb adatok esetén, vagy a szabályzattal kapcsolatos kérdéseivel forduljon Az OTDT Titkársághoz a titkar@otdt.hu címen.

Kérjük, minden esetben írja meg problémájának pontos voltát (lehetőség szerint küldjön képernyőképet is róla), az alábbi adatokat pontosan megadva:

- Azonosító (a pályamunka azonosítója)
- Cím (a pályamunka címe)
- Kapcsolattartó (a pályamunkát regisztráló személy neve, e-mail címe)
- intézmény, kar, konferencia (mely intézmény mely karának mely konferenciáján vett részt, a pontos dátumot megadva)

ADATAIM MENÜPONT

A menüpontra kattintva a következő al-menükből választhat:



51. ábra. Az "Adataim" menü

ADATOK MÓDOSÍTÁSA

Az „Adatok módosítása” menüpontra kattintva a lenyíló panelon szerkesztheti adatait a regisztrációnál már megismert módon.

JELSZÓ MÓDOSÍTÁSA

A „Jelszó módosítása” menüpontra kattintva a lenyíló panelon lehetősége nyílik jelszavának megváltoztatására.

 A screenshot of a web form titled 'Jelszó megváltoztatása'. The form has a yellow background and a red header. It contains three input fields labeled 'Régi jelszó:', 'Új jelszó:', and 'Megerősítés:'. Below the fields are two buttons: 'Mégsem' and 'Megváltoztat'.

52. ábra. Jelszó megváltoztatása

PÁLYAMUNKÁIM

A „Pályamunkáim” menüpontra kattintva a lenyíló panelon regisztrált és nevezett pályamunkáinak adatait tekintheti, illetve ellenőrizheti, hogy nevezését az OTDT titkárság és a rendezők elfogadták-e.

Pályamunka kiválasztásához kattintson annak címre.

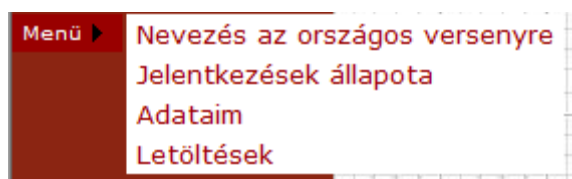
LETÖLTÉSEK MENÜPONT

Ezen menüpontból letölthetőek az esetlegesen fontos dokumentumok, nevezési lap(ok), kérvények, stb.

Még fejlesztés alatt.

TDK FELELŐSÖK

Bejelentkezés után a menüből egyelőre négy menüpont elérhető.



53. ábra. A TDK felősök által elérhető menüpontok

AZ „ADATAIM” MENÜPONT

Az „**Adataim**” pont alatt szerkeszthetik személyes adataikat, illetve itt van lehetőség a jelszó megváltoztatására a megfelelő linkre kattintva.

Az **e-mail** mező szerkesztésére nincs lehetőség, mivel ez a felhasználónév is egyben.

A **Teljes név**, az **Anyja neve**, illetve a **megjegyzés** mező egyszerű beírással szerkeszthető.

A **Születési Dátum** megadására a mező melletti **naptár ikonra** kattintva van lehetőség. Ebből kattintással kiválasztható a megfelelő dátum. A gyorsabb navigálás érdekében kattintson a hónap nevére, ekkor az adott év hónapjaiból választhat. Ha ekkor az évszámba kattint, akkor évtizedet választhat.



54. ábra. A naptár használata

Az **Intézmény** megadására a mező melletti  gombra kattintva van lehetőség.

Ekkor a megjelenő ablakban az **intézmény nevére kattintva** választhatja az ki. Egy oldalon tíz intézmény jelenik meg. **Lapozásra** a lista alján található számokra kattintva van lehetőség.

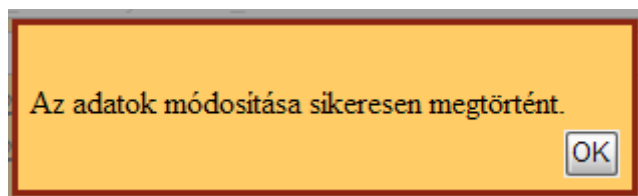
Választását a **Kiválasztás** gombra kattintva véglegesítheti, ekkor az intézmény neve megjelenik az előző lapon.

Intézmény	Felelős	Rövidítés	Város	aktív
Intézményen kívüli	Szatmári András	-	-	<input checked="" type="checkbox"/>
A Tan Kapuja Buddhista Főiskola	Farkas Attila Márton	TKBF	Budapest	<input checked="" type="checkbox"/>
Általános Vállalkozási Főiskola	Dr. Szalai Ibolya	ÁVF	Budapest	<input checked="" type="checkbox"/>
Apor Vilmos Katolikus Főiskola	Kádár Gyula	AVKF	Vác	<input checked="" type="checkbox"/>
Budapesti Corvinus Egyetem	Dr. Pénzes Béla	BCE	Budapest	<input checked="" type="checkbox"/>
Budapesti Gazdasági Főiskola	Dr. Harmat Zsigmond	BGF	Budapest	<input checked="" type="checkbox"/>
Budapesti Kommunikációs és Üzleti Főiskola	Dr. Róka Jolán	BKF	Budapest	<input checked="" type="checkbox"/>
Budapesti Műszaki és Gazdaságtudományi Egyetem	Dr. Kovács Ádám	BME	Budapest	<input checked="" type="checkbox"/>
Budapesti Műszaki Főiskola	Dr. Borbély Endre	BMF	Budapest	<input checked="" type="checkbox"/>
Debreceni Egyetem		DE	Debrecen	<input checked="" type="checkbox"/>
1 2 3 4 5 6				
<input type="text"/> <input type="button" value="Keresés"/> <input type="button" value="Kiválasztás"/>				

55. ábra. Az intézmény kereső használata

A megjegyzés mezőn kívül minden kitöltése kötelező. Amennyiben nem kívánja megadni születési dátumát, és anyjának nevét, úgy kérjük, a dátum esetén válassza az aktuális dátumot, az anyja nevéhez pedig a „nincs adat” feliratot írja. (Ezt a legtöbb esetben az OTDT titkárság már megtette).

Adatait a **mentés** gombra kattintva mentheti el. Sikeres mentés esetén „**Az adatok módosítása sikeresen megtörtént.**” üzenetet kapja. Hiányzó adatra mentéskor a mező mellett megjelenő piros csillag figyelmeztet.



56. ábra. Üzenet a sikeres módosításról

Amennyiben adataiban további hiányosságokat tapasztalt, illetve azok tévesek, kérjük ezt mindenképpen jelezze az OTDT titkárság felé a titkar@otdt.hu illetve az otdkadmin@gmail.com címen.

A „NEVEZÉS AZ ORSZÁGOS VERSENYRE” MENÜPONT.

Ez alatt a menüpont alatt vehetnek fel, illetve módosíthatják a helyi versenyek **jegyzőkönyveit**, illetve tekinthetik meg a már **regisztrált pályamunkákat**.

Első lépésként a legördülő listából válassza azt az intézményt, illetve kart, melynek jegyzőkönyveit látni szeretné (ez a legtöbb esetben egy-egy). Ha az intézményhez nem tartozik kar, válassza a **Nincs kar** lehetőséget.

Amennyiben a listában nem található meg az intézmény illetve kar, ahol ön TDK felelős, ezt mindenképpen jelezze a titkar@otdt.hu címen illetve az otdkadmin@gmail.com címen.

Intézmény:	Debreceni Egyetem	
Kar:	Informatikai Kar	
Kezdődátum	Végdátum	Elnök
2008:11:11	2008:11:20	Dr. Végh János

57. ábra. Jegyzőkönyvek

A megfelelő intézményt és kart kiválasztva egy listában megjelennek az ahhoz tartozó jegyzőkönyvek. Egy jegyzőkönyvet kiválasztva megjelennek annak részletes adatai, illetve a **Pályamunkák** link. Erre a linkre kattintva tekintheti meg a jegyzőkönyvhöz tartozó, már regisztrált pályamunkák adatait. Erről részletesen később.

Intézmény:	Debreceni Egyetem	
Kar:	Informatikai Kar	
Kezdődátum	Végdátum	Elnök
2008:11:11	2008:11:20	Dr. Végh János

[Pályamunkák](#)

58. ábra. Jegyzőkönyv kiválasztása

Az **Új gombra** kattintva van lehetősége új jegyzőkönyv adatainak rögzítésére. Ekkor a gomb felirata Mentésre változik.

Új	Módosítás
----	-----------

Mentés	Módosítás	Mégsem
--------	-----------	--------

59. ábra. A gombok használata

Az **iktatószámot** a **Titkárság** tölti ki, ez alapesetben **0000/000**.

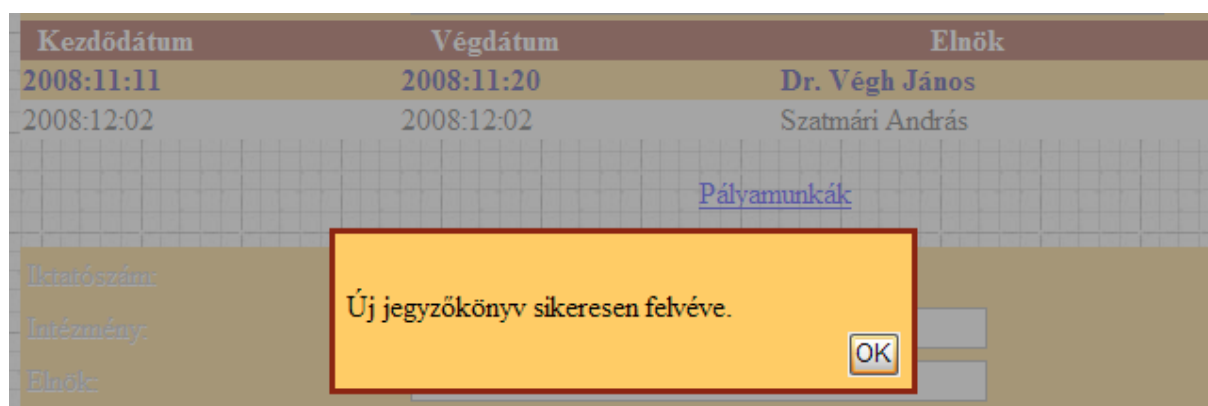
Az **Intézmény**, illetve **Elnök** mezők módosítására nincs lehetőség. Az elnök nevéként alapértelmezetten a saját nevét (a bejelentkezett felhasználó nevét) kell látnia.

Az **Időpont** mezők a konferencia időpontjának megadására szolgálnak. (**Kezdődátum – Végdátum formában**). Amennyiben nem többnapos volt a rendezvény, kérjük, mindkét dátumhoz ugyanazt adja meg. Szerkesztésük a naptár ikonokra kattintva lehetséges.

Naptár használata: kattintással kiválasztható a megfelelő dátum. A gyorsabb navigálás érdekében kattintson a hónap nevére, ekkor az adott év hónapjaiból választhat. Ha ekkor az évszámra kattint, akkor évtizedet választhat.

A **Mentésre** kattintva mentheti el a megadott adatokat. A sikeres mentésről értesítést kap: „**Új jegyzőkönyv sikeresen felvéve.**” és a gomb felirata **Újra** változik, illetve az újonnan felvett

jegyzőkönyv megjelenik a jegyzőkönyvek listájában. Hibás, vagy hiányzó adat esetén ez nem történik meg, ezekre a hibás mező mellett megjelenő **piros csillag** vagy üzenet figyelmeztet.



60. ábra. Üzenet az új jegyzőkönyv felvételének sikerességéről

A **Módosítás** gombra kattintva a listából éppen kiválasztott jegyzőkönyv adatait szerkesztheti az **Új gomb** használatánál már megismert módon.

PÁLYAMUNKÁK

A képeken is látható Pályamunkák linkre kattintva tekintheti meg a jegyzőkönyvhöz már regisztrált, illetve nevezett pályamunkákat.

Ez ugyanaz a felület, amit a hallgatók is használnak, annyi különbséggel, hogy ők csak a saját pályamunkájukat (vagy pályamunkáikat) látják, míg a felelősök az összeset.

Használatának részletes leírása megtalálható a hallgatóknak szóló fejezetben, vagy bejelentkezés után ezen a linken: http://otdk.unideb.hu/pdf/docs/Hasznalati_utmutato.pdf

A működésben annyi eltérés van, hogy amíg hallgató esetén a pályamunka kapcsolattartójaként automatikusan a pályamunkát regisztráló hallgató jelenik meg, addig itt meg kell adni azt a „...” vagy az Új gombra kattintva a kapcsolattartó mező mellett.

A ... gomb esetén a rendszerbe már regisztrált hallgatót választhat kapcsolattartónak, hasonló módon, mint ahogy az intézményeknél már. Az Új gomb esetén Új hallgatót regisztrálhat a felugró ablakban. A regisztrációról a hallgató e-mailben értesítést kap.

Az országos versenyre már jelentkezett pályamunkák esetén az Alap adatok fül alján a „Jelentkezési lap nyomtatása” gomb jelenik meg. Még nem nevezett pályamunkáknál ugyanezen a helyen a „Jelentkezés az országos versenyre” link.

Az országos versenyre már nevezett pályamunkák módosítására egyelőre nincs lehetőség.

JELENTKEZÉSEK ÁLLAPOTA MENÜPONT

A TDK felelősök ezen menüpont alatt tekinthetik meg a saját intézményükből, illetve karukról beérkezett nevezéseket, azoknak állapotát, az esetlegesen szükséges hiánypótlásokat.

Amennyiben a bejelentkezett felhasználó Intézményi felelős, úgy intézményének összes karának összes dolgozatát megtekintheti.

Amennyiben a bejelentkezett felhasználó több intézménynél, és/vagy több karnál is TDK felelős, úgy azoknak összes pályamunkáját megtekintheti.

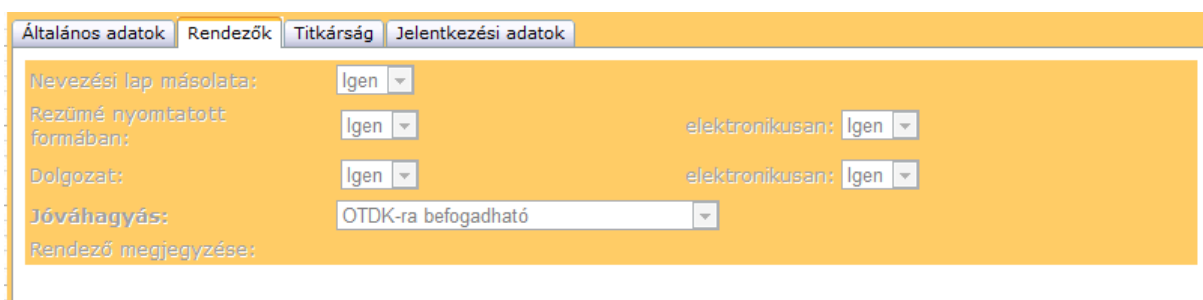
Intézmény és kar választásához használja az oldal tetején található két legördülő listát. A dolgozatok adatait a „Listáz” gombra kattintva tekintheti meg.

The screenshot shows a web interface with two dropdown menus. The first dropdown menu is set to 'Debreceni Egyetem' and the second is set to 'Informatikai Kar'. Below the dropdowns are two buttons: 'Listáz' and 'Nyomtatható változat'.

61. ábra. Intézmény és kar kiválasztása a nevezett dolgozatok megtekintéséhez

A „Nyomtatható változat” gombra kattintva letölthet egy rövidített összefoglalót a dolgozatokról (.pdf formátumban).

Egy dolgozat részletes adatainak megtekintéséhez kattintson a dolgozat címére.

The screenshot shows a form with four tabs: 'Általános adatok', 'Rendezők', 'Titkárság', and 'Jelentkezési adatok'. The 'Általános adatok' tab is active. It contains several fields: 'Nevezési lap másolata:' with a dropdown set to 'Igen'; 'Rézümé nyomtatott formában:' with a dropdown set to 'Igen'; 'Dolgozat:' with a dropdown set to 'Igen'; 'Jóváhagyás:' with a dropdown set to 'OTDK-ra befogadható'; 'Rendező megjegyzése:'; 'elektronikusan:' with a dropdown set to 'Igen'; and another 'elektronikusan:' with a dropdown set to 'Igen'.

62. ábra. A kiválasztott dolgozat adatai

LETÖLTÉSEK MENÜPONT

Ezen menüpontból letölthetőek az esetlegesen fontos dokumentumok, nevezési lap(ok), kérvények stb.

Még fejlesztés alatt.

HIBAJELENTÉS

Tudjuk és tapasztaljuk azt, hogy ezen lehetőségek még korántsem fedik le a TDK felelősök munkájának minden részletét, de rendszerünket folyamatosan építjük és tökéletesítjük. A rendszer fejlesztése során folyamatosan fognak megjelenni újabb lehetőségek és menüpontok.

A már elkészült felületeken is lehetnek, sőt biztos, hogy vannak is hibák, ezeket folyamatosan javítjuk, ezért kérjük, ha ilyesmivel találkozunk, azt megértéssel fogadják, és jelezzék a titkar@otdt.hu és az otdkadmin@gmail.com címeken.

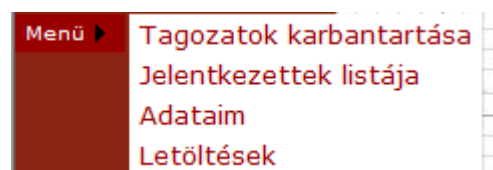
Hiba jelzésekor kérjük, pontosan adják meg az alábbi adatokat:

- Felhasználó neve és e-mail címe
- Intézmény, kar
- Jegyzőkönyv dátuma (ha a jegyzőkönyvnél jelentkezne probléma)
- Pályamunka címe (ha pályamunkánál jelentkezne probléma)
- A hiba részletes leírása: mikor, hol, milyen művelet után jelentkezett
- Lehetőség szerint képernyőkép a hibáról

ORSZÁGOS SZEKCIÓ RENDEZŐI

Bejelentkezés után az alábbi négy menüpontból választhatnak:

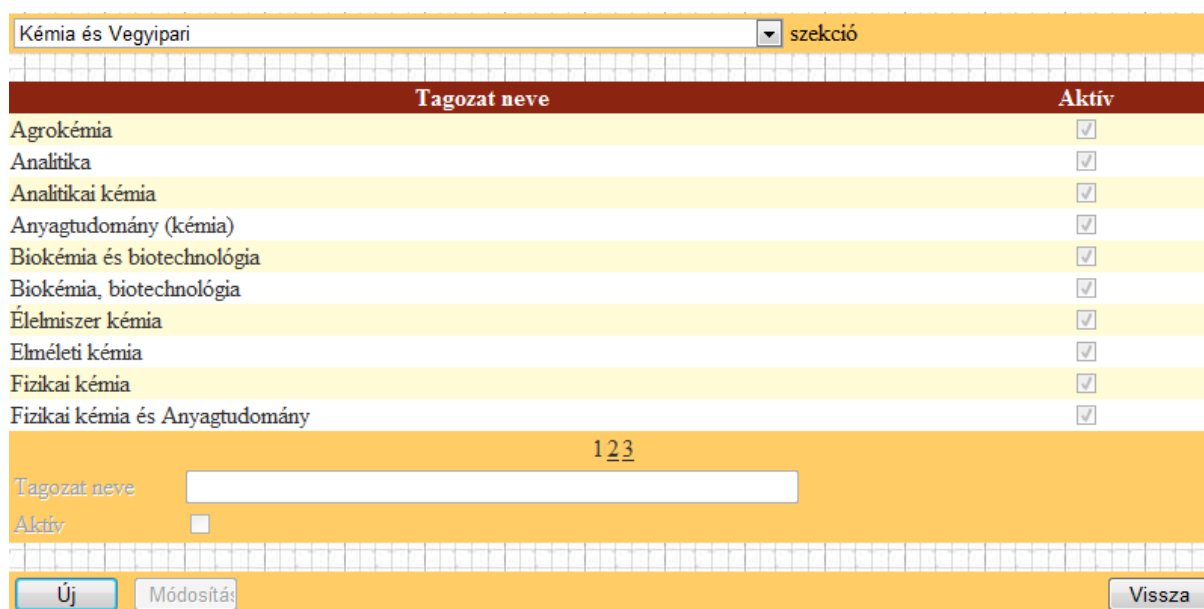
- Tagozatok karbantartása
- Jelentkezettek listája
- Adataim
- Letöltések



63. ábra. A szekció rendezők számára elérhető menüpontok

TAGOZATOK KARBANTARTÁSA

A „Tagozatok karbantartása” oldal lehetőséget biztosít az adott szekción belül új tagozatok létrehozására, és módosítására. Tagozat törlésére nincs lehetőség. Egy tagozatot aktívvá, vagy passzívvá lehet tenni. Csak aktív tagozatra jelentkezhet hallgató.

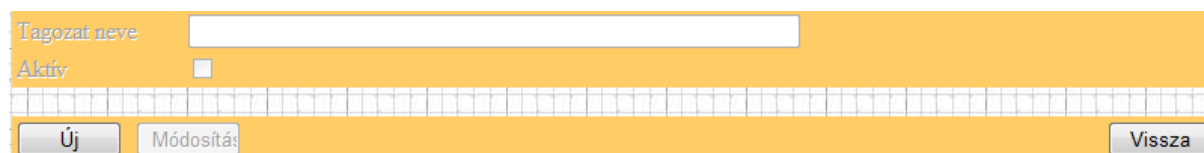


Tagozat neve	Aktív
Agrokémia	<input checked="" type="checkbox"/>
Analitika	<input checked="" type="checkbox"/>
Analitikai kémia	<input checked="" type="checkbox"/>
Anyagtudomány (kémia)	<input checked="" type="checkbox"/>
Biokémia és biotechnológia	<input checked="" type="checkbox"/>
Biokémia, biotechnológia	<input checked="" type="checkbox"/>
Élelmiszer kémia	<input checked="" type="checkbox"/>
Elméleti kémia	<input checked="" type="checkbox"/>
Fizikai kémia	<input checked="" type="checkbox"/>
Fizikai kémia és Anyagtudomány	<input checked="" type="checkbox"/>

64. ábra. Tagozatok karbantartása

Az itt látható listában a lista alján található oldalszámokra kattintva lapozhat.

ÚJ TAGOZAT FELVÉTELE



65. ábra. Új tagozat felvétele - a mezők inaktív állapotban

Új tagozat felvételéhez kattintson az „Új” gombra. A gomb felirata ekkor „Mentés”-re változik.

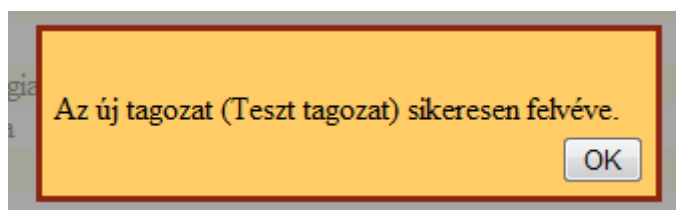
Tagozat neve

Aktív ☒

66. ábra. Új tagozat felvétele - a mezők szerkesztése

A „Mentés” gombra kattintva mentheti el az újonnan létrehozott tagozatot. Ezután az megjelenik a tagozatok listájában. Lásd fentebb.

A sikeresen végrehajtott mentésről üzenetet kapunk.



67. ábra. Üzenet az új tagozat sikeres felvételéről

TAGOZAT MÓDOSÍTÁSA

Már meglévő tagozat módosításához kattintson a módosítandó tagozat nevére, majd a módosítás gombra kattintva szerkesztheti azokat.

A gomb működése megegyezik az „Új tagozat felvétele” pontban leírt „Új gomb” működésével.

JELENTKEZETTEK LISTÁJA

Ezen az oldalon tekintheti meg a szekciójába nevezett dolgozatokat.

Kémia és Vegyipari

Jelentkezések exportálható listája

Összefoglaló nyomtatása

Pályamunka címe	Egyetem	OTDT jóváhagyás	Rendező jóváhagyás
1,4-Diszubsztituált kubánvegyületek ezüstionok hatására lejátszódó átrendeződési reakciói	ELTE TTK	i	i
1-Nitro-ciklohexén szelektív katalitikus hidrogénezésének vizsgálata nagynyomású áramlásos rendszerben	ELTE TTK	i	i
2,5-Diszubsztituált-1,3,4-oxadiazolok előállítása és átalakításai	DE TTK	i	i
2-Alkil-3-azidokromanonok szintézise és átalakításuk 2-alkil-3-aminokromonokká	DE TTK	i	i
2-Benzilidén-6-metoxi-1-tetralonok szintézise	SZTE TTIK	i	i
5-HT6 receptoron ható új vegyületek szintézise	BME VBK	i	i
A 3 Dimenziós molekulaterve és automatizálása egy webes portálrendszeren keresztül	SZTE JGYPK	i	i
A 4-klórfenol átalakítása ftalocianin katalizátorokkal	SZTE TTIK	i	i
A bizmut-tellurid elektrokémiai leválasztása és tulajdonságai	SZTE TTIK	i	i
A CH3OOH + OH reakció kinetikájának elméleti vizsgálata	ELTE TTK	i	i

12345678910...

Keresés

Összes

68. ábra. Jelentkezettek listája

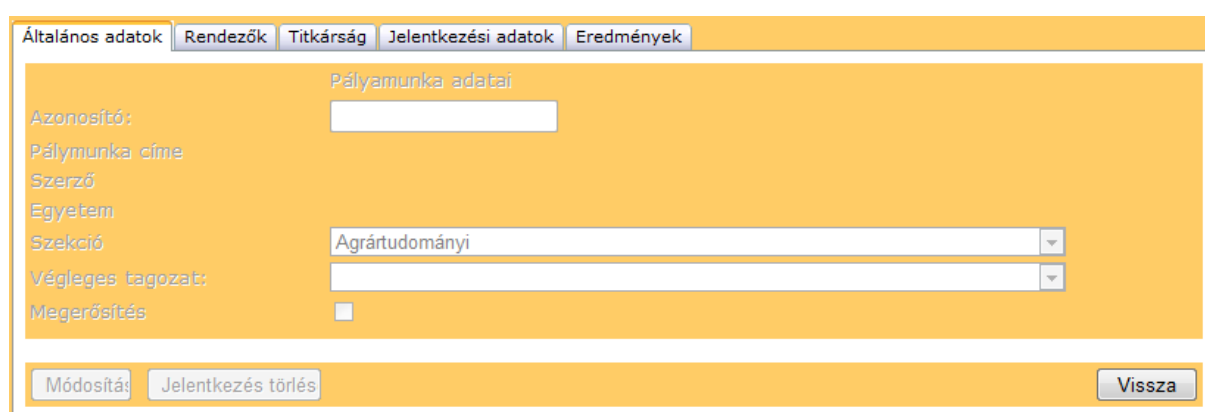
Az „Összefoglaló nyomtatása” gombra kattintva letölthet egy rövidített összesítőt a szekciójába nevezett dolgozatokról, és azoknak állapotáról (szükséges hiánypótlások, stb). A fájl formátuma .pdf.

A „Jelentkezések exportálható listája” menüpontra kattintva egy táblázatkezelőbe (pl MS Excel) másolható táblázatot kap a jelentkezések részletes adataival. A funkció még fejlesztés alatt.

Jelmagyarázat: i – a pályamunka befogadható, n - nem befogadható, h - hiánypótlás szükséges.

A pályamunka, vagy dolgozat részletes adatait a címére kattintva tekintheti meg az itt látható táblázatban.

ÁLTALÁNOS ADATOK FÜL



69. ábra. Általános adatok

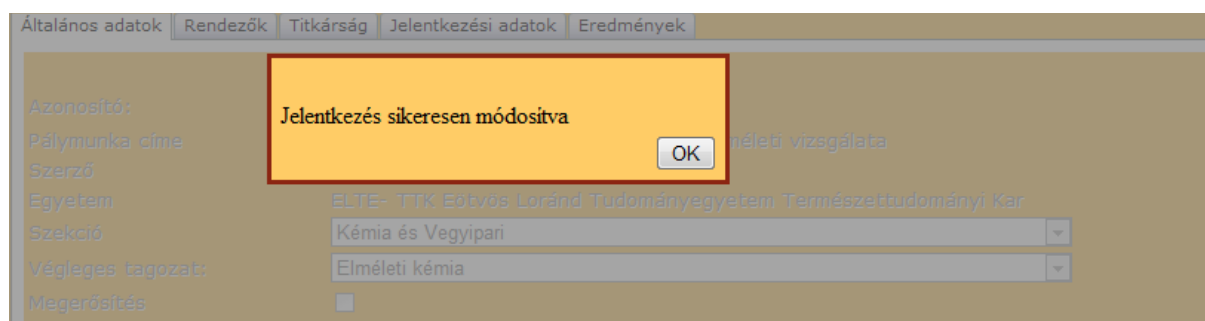
Ezen a fülön tekintheti meg a pályamunka adatainak rövid összefoglalóját.

A pályamunka részletes adatainak megtekintéséhez kattintson a „Pályamunka adatai” linkre. A megnyíló ablakban a pályamunka adatait a Hallgatók valamint a TDK felelősök fejezetben leírt módon tekintheti meg, azok szerkesztéséhez nincs joga.

A „Módosítás” gombra kattintva a gomb felirata „Mentés”-re változik és a „Végleges tagozat” legördülő listából választva lehetősége van a pályamunkát egy másik tagozatba átsorolni.

A változtatásokat a „Mentés” gombra kattintva véglegesíti.

A mentés sikerességéről üzenet tájékoztat.

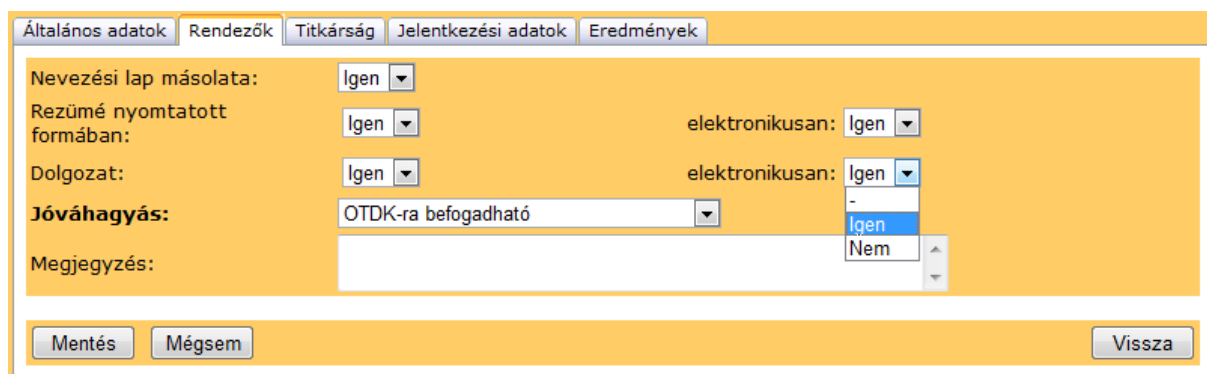


70. ábra. Üzenet a sikeres módosításról

RENDEZŐK FÜL

Ezen a fülön regisztrálhatja a jelentkezés elfogadásához szükséges dokumentumok beérkezését, vagy be nem érkezését, illetve, hogy a pályamunka az OTDK-ra befogadható-e vagy sem.

A „Módosítás” gombra kattintva a gomb felirata „Mentés”-re változik és szerkeszthetővé válnak a mezők.

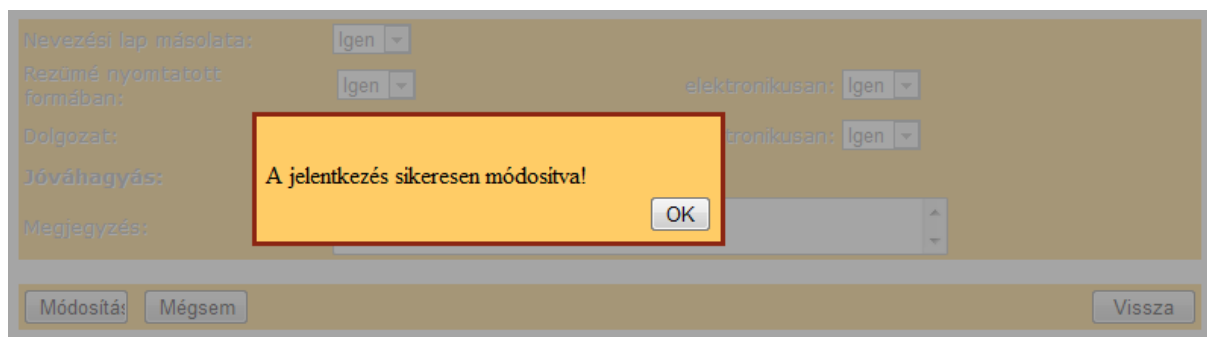


71. ábra. Rendezők fül

A változtatásokat a „Mentés” gombra kattintva véglegesíti.

A változtatásokat a „Mégsem” gombbal elvetheti.

A mentés sikerességéről üzenet tájékoztat.



72. ábra. Üzenet a sikeres módosításról

TITKÁRSÁG FÜL

Ezen a fülön az OTDT titkárság regisztrálja a jelentkezés elfogadásához szükséges dokumentumok beérkezését, vagy be nem érkezését, illetve, hogy a pályamunka az OTDK-ra befogadható-e vagy sem.

Ezen a fülön nincs joga az adatok szerkesztéséhez.

Általános adatok	Rendezők	Titkárság	Jelentkezési adatok	Eredmények
Iktatószám:	OTDT-B-2008/945			
Azonosító:	08/116			
Nevezési lap:	Igen ▼			
Rezumé:	Igen ▼			
Tartalmi címazonosság:	Igen ▼			
Nyilatkozat(végzettség esetén):	- ▼			
Összesítő jegyzék:	Igen ▼			
Hallgató aláírása:	Igen ▼			
Jóváhagyás:	OTDK-ra befogadható ▼			
Megjegyzés:	A nevezési lapon nem jelölte, hogy első vagy másoddiplomás. Kérjük az adat pótlását. A kért			
Módosítás			Vissza	

73. ábra. A "Titkárság" fül

JELENTKEZÉSI ADATOK FÜL

Általános adatok	Rendezők	Titkárság	Jelentkezési adatok	Eredmények
Eredeti szekció:	Kémia és Vegyipari			
Legemegfelelőbb tagozat:	Reakciókinetika és katalízis			
Megfelelő tagozat:	Szerves kémia			
				Vissza

74. ábra. Jelentkezési adatok

Ezen a fülön tekintheti meg, hogy eredetileg melyik szekcióba és mely tagozatokba jelentkezett a pályamunka.

Ezen a fülön nincs joga az adatok szerkesztéséhez.

EREDMÉNYEK FÜL

Általános adatok	Rendezők	Titkárság	Jelentkezési adatok	Eredmények
Szekció:	Kémia és Vegyipari			
Tagozat:	Katalízis ▼			Új tagozat
Nem bemutatott dolgozat	<input type="checkbox"/>			
Helyezés				
Különdíjban részesül	<input type="checkbox"/>			
Megjegyzések, kiegészítések:				
Mentés		Mégsem		

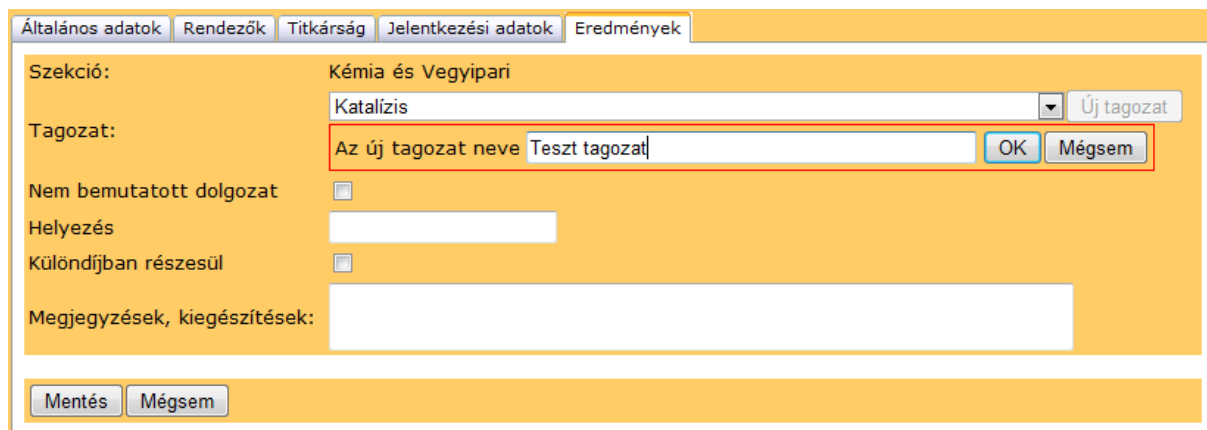
75. ábra. Eredmények

Ezen a fülön adhatja meg a nevezéshez a szekcióban elért eredményeket.

A „Módosítás” gombra kattintva a gomb felirata „Mentés”-re változik és szerkeszthetővé válnak a mezők.

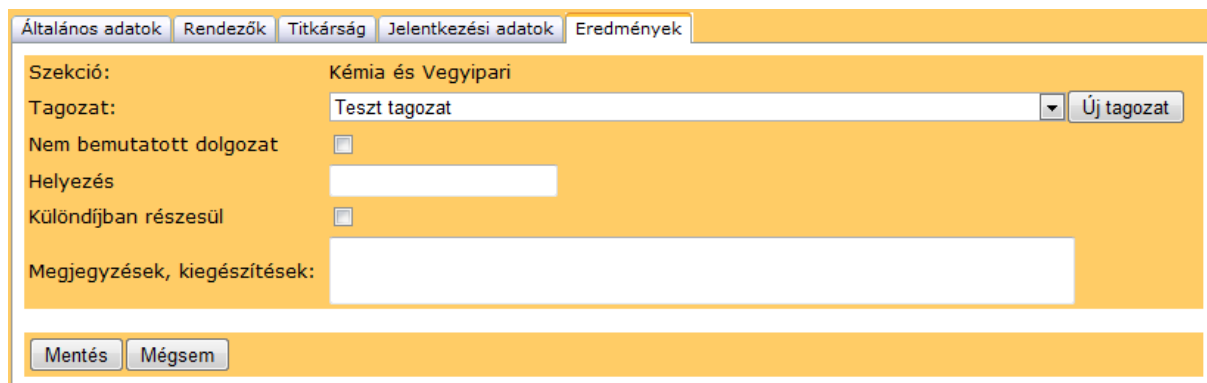
A „Tagozat” legördülő listából választva adhatja meg, hogy mely tagozatban indult a dolgozat. Amennyiben a megfelelő tagozat van kiválasztva, nincs szükség a tagozat módosítására. Amennyiben a keresett tagozat nem szerepel a listában az „Új tagozat” gombra kattintva veheti fel azt a megjelenő pirossal keretezett részben.

Megjegyzés: új tagozat felvételére lehetőség van a „Tagozatok karbantartása” menüpont alatt is, a fentebb már ismertetett módon.



76. ábra. Új tagozat felvétele az Eredmények fülön

Az OK gombra kattintva az új tagozat bekerül a tagozatok listájába, és automatikusan kiválasztásra kerül.



77. ábra. Sikeresen felvett új tagozat

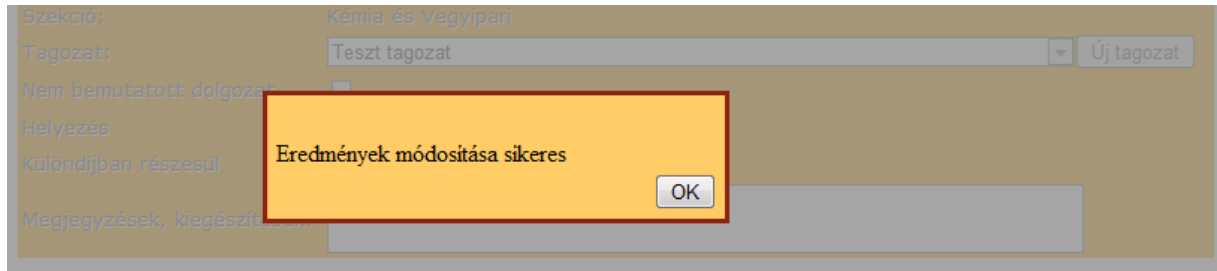
Amennyiben a dolgozat nem került bemutatásra a konferencián tegyen pipát a „Nem bemutatott dolgozat” mezőbe. Ez esetben a többi mező nem szerkeszthető.

A „Helyezés” mezőbe a konferencián elért helyezés kerül. A „Különdíjban részesül” mellé tegyen pipát amennyiben a hallgató különdíjat szerzett. A „Megjegyzések, kiegészítések” mezőbe rögzítheti a különdíj jellegét, egyéb megjegyzéseket.

A változtatásokat a „Mentés” gombra kattintva véglegesíti.

A változtatásokat a „Mégsem” gombbal elvetheti.

A mentés sikerességéről üzenet tájékoztat



78. ábra. Üzenet a sikeres módosításról

ADATAIM

Adatainak szerkesztésére a „TDK felelősök” fejezetben ismertetett módon van lehetőség.

LETÖLTÉSEK

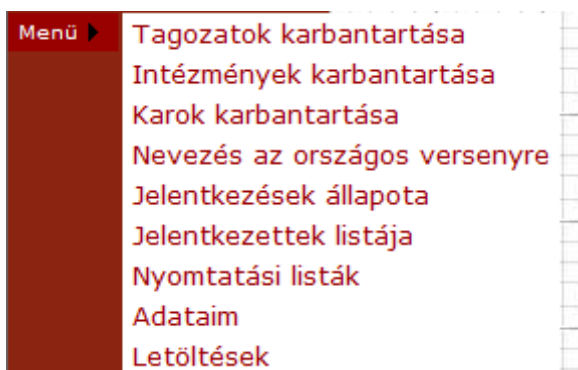
Ezen menüpontból letölthetőek az esetlegesen fontos dokumentumok, nevezési lap(ok), kérvények stb.

Még fejlesztés alatt.

OTDT TITKÁRSÁG

Bejelentkezés után az alábbi kilenc menüpontból választhatnak:

- Tagozatok karbantartása
- Intézmények karbantartása
- Karok karbantartása
- Nevezés az országos versenyre
- Jelentkezések állapota
- Jelentkezettek listája
- Nyomtatási listák
- Adataim
- Letöltések



79. ábra. Az OTDT titkársági tagok számára elérhető menüpontok

TAGOZATOK KARBANTARTÁSA

Leírását lásd „Országos szekció rendezői” fejezetben.

Az ott leírtakhoz képest eltérés, hogy az OTDT titkársági tag bármely szekcióba vehet fel új tagozatot, és bármely szekció bármely tagozatát szerkesztheti.

INTÉZMÉNYEK KARBANTARTÁSA

Röv.	Intézmény neve	aktív
GYHF	Győri Hittudományi Főiskola	<input checked="" type="checkbox"/>
HJF	Harsányi János Főiskola	<input checked="" type="checkbox"/>
HFF	Heller Farkas Gazdasági és Turisztikai Szolgáltatások Főiskolája	<input checked="" type="checkbox"/>
KE	Kaposvári Egyetem	<input checked="" type="checkbox"/>
KRE	Károli Gáspár Református Egyetem	<input checked="" type="checkbox"/>
KRF	Károly Róbert Főiskola	<input checked="" type="checkbox"/>
KTDK	Kárpátaljai Területi Tudományos Diákköri Konferencia	<input checked="" type="checkbox"/>
KF	Kecskeméti Főiskola	<input checked="" type="checkbox"/>
KJF	Kodolányi János Főiskola	<input checked="" type="checkbox"/>
KTIF	Kölcsey Ferenc Református Tanítóképző Főiskola	<input checked="" type="checkbox"/>

1234567

Intézmény neve

Aktív ☐

Virtuális ☐

Rövidítés

Irányítószám

Város

Utca, házszám

TDK felelős: ...

80. ábra. Intézmények karbantartása

A fenti listában valamelyik intézményre kattintva választhatjuk ki azt, hogy lássuk annak adatait.

KERESÉS

A keresés mezőbe szöveget írva, majd a keresés gombot megnyomva szűkíthető az intézmények listája. Ez hasznos lesz, ha már több száz intézmény között kell keresgélni.



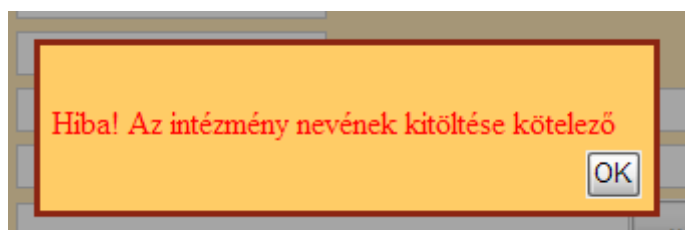
81. ábra. A keresés használata

Az összes intézmény úgy jeleníthető meg újra, ha töröljük a keresés mező tartalmát, és megnyomjuk a keresés gombot.

ÚJ INTÉZMÉNY FELVÉTELE

Az „Új” gombra kattintva lehet új intézményeket felvenni, ilyenkor az eddig inaktív szövegmezők aktívvá válnak (kivétel a Felelős neve), valamint a listában nem lesz kiválasztott intézmény, illetve az Új gomb felirata „Mentés”-re változik.

Itt lehet megadni az Intézmény adatait, minden mező kitöltése kötelező, ha valami üresen marad, akkor arra figyelmeztet a rendszer.



82. ábra. Hibaüzenet sikertelen mentésről

A TDK-felelős neve melletti mező inaktív marad, ez azért van, mert a felelőst más lapokon kezeljük.

A felelős neve melletti „...” gombra kattintva a megjelenő ablakban választhatunk Intézményi felelőst mindazon felhasználók közül, akik már jelen vannak a rendszerben és a szerepkörük megfelelő.

A felelőst a felugró ablakban a nevére kattintva választhatjuk ki, majd a jobb alsó sarokban lévő „Kiválasztás” gombra kattintva visszajutunk az előző lapra, ahol a felelős neve megjelenik.

Új felelős felvételéhez kattintsunk az „Új gombra”. Szintén új ablak jelenik meg, ahol megadhatjuk az új felhasználó adatait. A felhasználó alapértelmezetten megkapja a TDK felelős szerepkört.

Itt az „OK” gomb megnyomásával hozhatjuk létre a felhasználót (az e-mail cím lesz egyben a felhasználó név is), amennyiben nem volt hiba kapunk egy „Felhasználó sikeresen létrejött” üzenetet (ha nem akkor a hibaüzenetet), és az „OK” gomb felirata „Bezár/Felvesz”-re változik. Erre a gombra kattintva az előző lapra, ahol a felelős neve megjelenik. (FONTOS: Ha az ablakot nem ezzel a gombbal zárjuk be, a felhasználó, akkor is létrejön, de adatai nem rendelődnek az intézményhez).



The screenshot shows a user creation form with the following fields and options:

- E-mail:
- Teljes név:
- Születési dátum: (with a calendar icon)
- Anyja neve:
- Megjegyzés:

On the right side, there is a list of roles with checkboxes:

- ☐ Admin
- ☐ Hallgató
- ☒ Helyi TDT elnök
- ☐ Kari TDK felelős
- ☐ Országos szekció rendezői
- ☐ OTDK bíráló
- ☐ OTDT titkárság

An "OK" button is located at the bottom right of the form.

83. ábra. TDK felelős felvétele az adott intézményhez

Miután visszatértünk az intézmények oldalára a „Mentés” gombra kattintva menthetjük el az elvégzett változtatásokat (ha valami hiba volt, akkor a hibaüzenet tájékoztat, ha nem volt hiba, akkor értesít a sikeres műveletről).

A „Mégse” gombra kattintva félbeszakíthatjuk a műveletet, de ha már hoztunk létre felhasználót, az akkor is megmarad.

INTÉZMÉNY MÓDOSÍTÁSA

Az intézmény módosítását a fentihez teljesen hasonló módon végezhetjük el a „Módosítás” gombra kattintva. Ekkor a listában éppen kiválasztott intézmény adatait tudjuk szerkeszteni a már leírt módon. Amennyiben nincs kiválasztott intézmény, nem is tudunk a „Módosítás” gombra kattintani, az inaktív, ellentétben az „Új” gombbal, hiszen intézményt bármikor vehetünk fel, amikor éppen nem egy másikat módosítunk.

KAROK KARBANTARTÁSA

Használata teljes mértékben megegyezik az Intézmények karbantartásánál megismerttel. Az eltérés csupán annyi, hogy mielőtt kart vehetnénk fel, vagy módosítanánk, ki kell választanunk azt az intézményt, amelyhez tartozó kart szerkeszteni szeretnénk. A kiválasztás az intézmény nevére kattintva lehetséges.

Amennyiben az intézményhez nem tartozik kar, akkor vegyünk fel egy „Nincs kar” nevű kart.

Röv.	Intézmény neve	aktív
-	Intézményen kívüli	<input checked="" type="checkbox"/>
TKBF	A Tan Kapuja Buddhista Főiskola	<input checked="" type="checkbox"/>
ÁVF	Általános Vállalkozási Főiskola	<input checked="" type="checkbox"/>
AVKF	Apor Vilmos Katolikus Főiskola	<input checked="" type="checkbox"/>
BCE	Budapesti Corvinus Egyetem	<input checked="" type="checkbox"/>
BGF	Budapesti Gazdasági Főiskola	<input checked="" type="checkbox"/>
BKF	Budapesti Kommunikációs és Üzleti Főiskola	<input checked="" type="checkbox"/>
BME	Budapesti Műszaki és Gazdaságtudományi Egyetem	<input checked="" type="checkbox"/>
BMF	Budapesti Műszaki Főiskola	<input checked="" type="checkbox"/>
DE	Debreceni Egyetem	<input checked="" type="checkbox"/>

1 2 3 4 5 6 7

Röv.	Kar neve	aktív
ÉSZK	Építészmérnöki Kar	<input checked="" type="checkbox"/>
ÉÖK	Építőmérnöki Kar	<input checked="" type="checkbox"/>
GTK	Gazdaság- és Társadalomtudományi Kar	<input checked="" type="checkbox"/>
GÉK	Gépészmérnöki Kar	<input checked="" type="checkbox"/>
KSK	Közlekedésmérnöki Kar	<input checked="" type="checkbox"/>
TTK	Természettudományi Kar	<input checked="" type="checkbox"/>
VBK	Vegyészmérnöki és Biomérnöki Kar	<input checked="" type="checkbox"/>
VIK	Villamosmérnöki és Informatikai Kar	<input checked="" type="checkbox"/>

Kar neve

Aktív ☐

Rövidítés

Kari felelős

84. ábra. Karok karbantartása

NEVEZÉS AZ ORSZÁGOS VERSENYRE

Működése megegyezik a „TDK felelősök” fejezetben már ismertetettel.

Eltérés, hogy az OTDT titkársági tag minden intézmény minden karának összes jegyzőkönyvét és összes pályamunkáját láthatja és szerkesztheti, valamint törölheti, ha még nincs hivatkozás rá.

JELENTKEZÉSEK ÁLLAPOTA

Működése megegyezik a „TDK felelősök” fejezetben már ismertetettel.

Eltérés, hogy az OTDT titkársági tag minden intézmény minden karának összes pályamunkáját láthatja.

JELENTKEZETTEK LISTÁJA

Ezen az oldalon tekintheti meg az összes szekcióba nevezett összes dolgozatot.

Kémia és Vegyipari

Jelentkezések exportálható listája

Összefoglaló nyomtatása

Pályamunka címe	Egyetem	OTDT jóváhagyás	Rendező jóváhagyás
1,4-Diszubsztituált kubánvegyületek ezüstionok hatására lejátszódó átrendeződési reakciói	ELTE TTK	i	i
1-Nitro-ciklohexén szelektív katalitikus hidrogénezésének vizsgálata nagynyomású áramlós rendszerben	ELTE TTK	i	i
2,5-Diszubsztituált-1,3,4-oxadiazolok előállítása és átalakításai	DE TTK	i	i
2-Alkil-3-azidokromanonok szintézise és átalakításuk 2-alkil-3-aminokromonokká	DE TTK	i	i
2-Benzilidén-6-metoxi-1-tetralonok szintézise	SZTE TTIK	i	i
5-HT6 receptoron ható új vegyületek szintézise	BME VBK	i	i
A 3 Dimenziós molekulaterv és automatizálása egy webes portálrendszeren keresztül	SZTE JGYPK	i	i
A 4-klórfenol átalakítása ftalocianin katalizátorokkal	SZTE TTIK	i	i
A bizmut-tellurid elektrokémiai leválasztása és tulajdonságai	SZTE TTIK	i	i
A CH3OOH + OH reakció kinetikájának elméleti vizsgálata	ELTE TTK	i	i

12345678910...

Keresés

Összes

85. ábra. Jelentkezettek listája

Az „Összefoglaló nyomtatása” gombra kattintva letölthet egy rövidített összesítőt a kiválasztott szekcióba nevezett dolgozatokról, és azoknak állapotáról (szükséges hiánypótlások, stb). A fájl formátuma .pdf.

A „Jelentkezések exportálható listája” menüpontra kattintva egy táblázatkezelőbe (pl MS Excel) másolható táblázatot kap a jelentkezések részletes adataival. A funkció még fejlesztés alatt.

Jelmagyarázat: i – a pályamunka befogadható, n - nem befogadható, h - hiánypótlás szükséges.

A pályamunka, vagy dolgozat részletes adatait a címére kattintva tekintheti meg az itt látható táblázatban.

ÁLTALÁNOS ADATOK FÜL

Általános adatok	Rendezők	Titkárság	Jelentkezési adatok	Eredmények
Pályamunka adatai				
Azonosító:	<input type="text"/>			
Pályamunka címe	<input type="text"/>			
Szerző	<input type="text"/>			
Egyetem	<input type="text"/>			
Szekció	Agrártudományi			
Végleges tagozat:	<input type="text"/>			
Megerősítés	<input type="checkbox"/>			
Módosítás	Jelentkezés törlés	Vissza		

86. ábra. Általános adatok fül

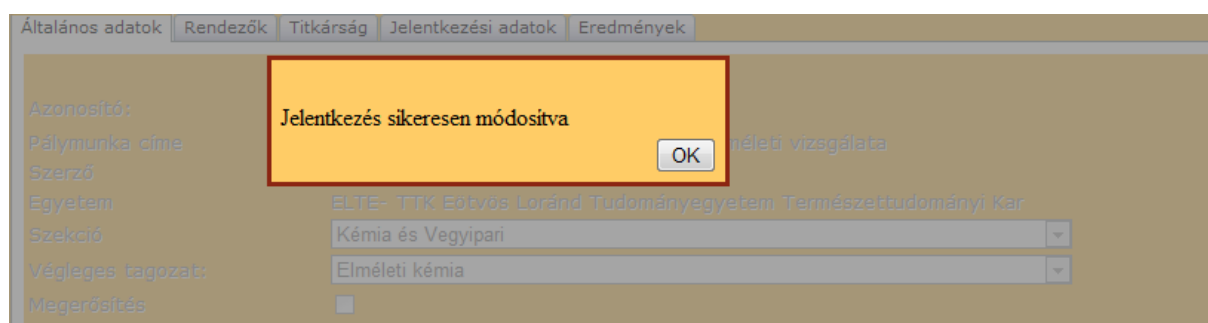
Ezen a fülön tekintheti meg a pályamunka adatainak rövid összefoglalóját.

A pályamunka részletes adatainak megtekintéséhez kattintson a „Pályamunka adatai” linkre. A megnyíló ablakban a pályamunka adatait a Hallgatók valamint a TDK felelősök fejezetben leírt módon tekintheti meg, azok szerkesztéséhez a titkársági tagnak van joga.

A „Módosítás” gombra kattintva a gomb felirata „Mentés”-re változik és a „Szekció” és „Végleges tagozat” legördülő listából választva lehetősége van a pályamunkát egy másik szekcióba és tagozatba átsorolni.

A változtatásokat a „Mentés” gombra kattintva véglegesíti.

A mentés sikerességéről üzenet tájékoztat.

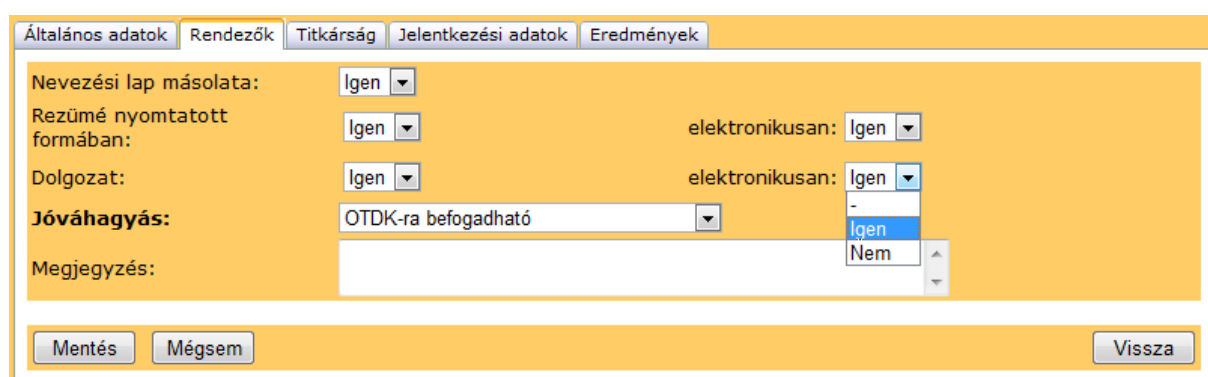
The screenshot shows a web application interface with a tabbed menu at the top: "Általános adatok", "Rendezők", "Titkárság", "Jelentkezési adatok", and "Eredmények". The "Jelentkezési adatok" tab is active. A yellow modal box is centered on the screen with the text "Jelentkezés sikeresen módosítva" and an "OK" button. In the background, the form fields for "Pályamunka címe", "Szerző", "Egyetem" (set to "ELTE- TTK Eötvös Loránd Tudományegyetem Természettudományi Kar"), "Szekció" (set to "Kémia és Vegyipari"), "Végleges tagozat" (set to "Elméleti kémia"), and "Megerősítés" (checkbox) are visible.

87. ábra. Üzenet a sikeres módosításról

RENDEZŐK FÜL

Ezen a fülön a rendezők regisztrálják a jelentkezés elfogadásához szükséges dokumentumok beérkezését, vagy be nem érkezését, illetve, hogy a pályamunka az OTDK-ra befogadható-e vagy sem.

A „Módosítás” gombra kattintva a gomb felirata „Mentés”-re változik és szerkeszthetővé válnak a mezők.

The screenshot shows the "Rendezők" tab selected in the application. The form contains several fields: "Nevezési lap másolata:" with a dropdown set to "Igen"; "Rézümé nyomtatott formában:" with a dropdown set to "Igen"; "Dolgozat:" with a dropdown set to "Igen"; "Jóváhagyás:" with a dropdown set to "OTDK-ra befogadható"; and "Megjegyzés:" with a text area. To the right of these fields are two "elektronikusan:" labels, each followed by a dropdown set to "Igen". At the bottom of the form are three buttons: "Mentés", "Mégsem", and "Vissza".

88. ábra. Rendezők fül

A változtatásokat a „Mentés” gombra kattintva véglegesíti.

A változtatásokat a „Mégsem” gombbal elvetheti.

A mentés sikerességéről üzenet tájékoztat.

89. ábra. Üzenet a sikeres módosításról

TITKÁRSÁG FÜL

Ezen a fülön az OTDT titkárság regisztrálja a jelentkezés elfogadásához szükséges dokumentumok beérkezését, vagy be nem érkezését, illetve, hogy a pályamunka az OTDK-ra befogadható-e vagy sem.

A „Módosítás” gombra kattintva a gomb felirata „Mentés”-re változik és szerkeszthetővé válnak a mezők.

90. ábra. Titkárság fül

A változtatásokat a „Mentés” gombra kattintva véglegesíti.

A változtatásokat a „Mégsem” gombbal elvetheti.

A mentés sikerességéről üzenet tájékoztat.

JELENTKEZÉSI ADATOK FÜL

Általános adatok Rendezők Titkárság **Jelentkezési adatok** Eredmények

Eredeti szekció: Kémia és Vegyipari

Legemelőbb tagozat: Reakciókinetika és katalízis

Megfelelő tagozat: Szerves kémia

Vissza

91. ábra. Jelentkezési adatok fül

Ezen a fülön tekintheti meg, hogy eredetileg melyik szekcióba és mely tagozatokba jelentkezett a pályamunka.

Ezen a fülön nincs joga az adatok szerkesztéséhez.

EREDMÉNYEK FÜL

Általános adatok Rendezők Titkárság **Jelentkezési adatok** **Eredmények**

Szekció: Kémia és Vegyipari

Tagozat: Katalízis

Nem bemutatott dolgozat ☐

Helyezés

Különdíjban részesül ☐

Megjegyzések, kiegészítések:

Mentés Mégsem

92. ábra. Eredmények fül

Ezen a fülön adhatja meg a nevezéshez a szekcióban elért eredményeket.

A „Módosítás” gombra kattintva a gomb felirata „Mentés”-re változik és szerkeszthetővé válnak a mezők.

A „Tagozat” legördülő listából választva adhatja meg, hogy mely tagozatban indult a dolgozat. Amennyiben a megfelelő tagozat van kiválasztva, nincs szükség a tagozat módosítására. Amennyiben a keresett tagozat nem szerepel a listában az „Új tagozat” gombra kattintva veheti fel azt a megjelenő pirossal keretezett részben.

Megjegyzés: új tagozat felvételére lehetőség van a „Tagozatok karbantartása” menüpont alatt is, a fentebb már ismertetett módon.

93. ábra. Új tagozat felvétele az eredmények fülön

Az OK gombra kattintva az új tagozat bekerül a tagozatok listájába, és automatikusan kiválasztásra kerül.

94. ábra. Sikeresen felvett új tagozat

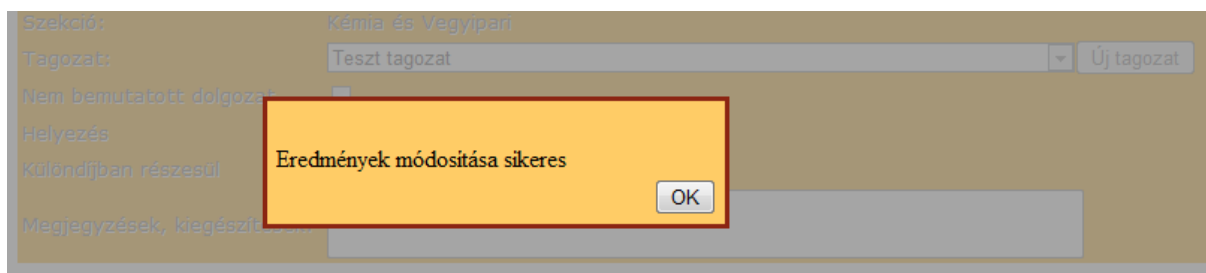
Amennyiben a dolgozat nem került bemutatásra a konferencián tegyen pipát a „Nem bemutatott dolgozat” mezőbe. Ez esetben a többi mező nem szerkeszthető.

A „Helyezés” mezőbe a konferencián elért helyezés kerül. A „Különdíjban részesül” mellé tegyen pipát amennyiben a hallgató különdíjat szerzett. A „Megjegyzések, kiegészítések” mezőbe rögzítheti a különdíj jellegét, egyéb megjegyzéseket.

A változtatásokat a „Mentés” gombra kattintva véglegesíti.

A változtatásokat a „Mégsem” gombbal elvetheti.

A mentés sikerességéről üzenet tájékoztat



95. ábra. Üzenet a sikeres mentésről

NYOMTATÁSI LISTÁK

Ezen az oldalon a jelentkezések állapotáról szóló rövidített összefoglaló(k) nyomtatására van lehetőség.

Még fejlesztés alatt.

96. ábra. Nyomtatási listák

Az intézmény neve előtti pipát kiválasztva egy intézményi összesítőt kapunk, karonként lebontva.

Ha intézményt és kart is választunk, az adott intézményből és karról nevezett összes dolgozatot láthatjuk szekciónkénti bontásban.

Ha intézményt, kart és szekciót is választunk az adott intézményből és karról az adott szekcióba nevezett összes dolgozatot láthatjuk

Ha csak szekciót választunk, az adott szekcióba nevezett összes dolgozatot kapjuk.

ADATAIM

Adatainak szerkesztésére a „TDK felelősök” fejezetben ismertetett módon van lehetőség.

LETÖLTÉSEK

Ezen menüpontból letölthetőek az esetlegesen fontos dokumentumok, nevezési lap(ok), kérvények stb.

Még fejlesztés alatt.

ADMINISZTRÁTOR (ADMIN)

Adminként természetesen az eddig említett összes művelet elvégezhető, valamint ezek a következő két ponttal bővülnek:

- Szekciók karbantartása
- Felhasználók karbantartása

SZEKCIÓK KARBANTARTÁSA

A „Szekciók karbantartása” oldal lehetőséget biztosít új szekciók létrehozására, és módosítására. Szekció törlésére nincs lehetőség. Egy szekciót aktívvá, vagy passzívvá lehet tenni. Csak aktív szekciókra jelentkezhet hallgató.

Szekció neve	aktív
-	<input type="checkbox"/>
Agrártudományi	<input checked="" type="checkbox"/>
Állam- és Jogtudományi	<input checked="" type="checkbox"/>
Biológia	<input checked="" type="checkbox"/>
enyim	<input checked="" type="checkbox"/>
Fizika, Földtudományok és Matematika	<input checked="" type="checkbox"/>
Had- és Rendészettudományi	<input checked="" type="checkbox"/>
Humán Tudományi (bölcsezet)	<input checked="" type="checkbox"/>
Informatika Tudományi	<input checked="" type="checkbox"/>
Kémia és Vegyipari	<input checked="" type="checkbox"/>

12

Szekció neve

Aktív ☐

Új Módosítás Vissza

97. ábra. Szekciók karbantartása

Az itt látható listában a lista alján található oldalszámokra kattintva lapozhat.

ÚJ SZEKCIÓ FELVÉTELE

Új szekció felvételéhez kattintson az „Új” gombra. A gomb felirata ekkor „Mentés”-re változik.

Szekció neve

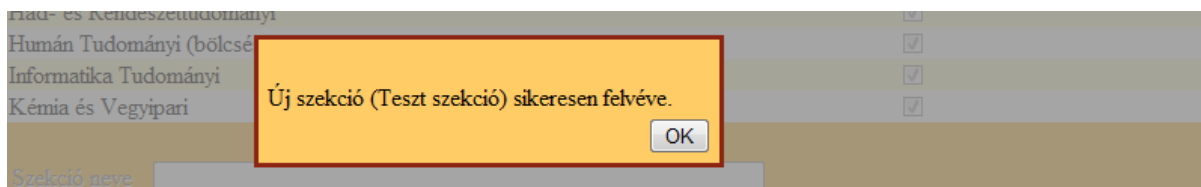
Aktív ☒

Mentés Módosítás Mégsem Vissza

98. ábra. Új szekció felvétele

A „Mentés” gombra kattintva mentheti el az újonnan létrehozott szekciót. Ezután az megjelenik a szekciók listájában. Lásd fentebb.

A sikeresen végrehajtott mentésről üzenetet kapunk.



99. ábra. Üzenet a mentés sikerességéről

SZEKCIÓ MÓDOSÍTÁSA

Már meglévő szekció módosításához kattintson a módosítandó szekció nevére, majd a módosítás gombra kattintva szerkesztheti azt.

A gomb működése megegyezik az „Új szekció felvétele” pontban leírt „Új gomb” működésével.

FELHASZNÁLÓK KARBANTARTÁSA

A felhasználók listájáról – mivel az személyes adatokat tartalmaz – itt képet nem teszünk közé. A lista felépítése megegyezik az eddig már ismertettekkel.

ÚJ FELHASZNÁLÓ LÉTREHOZÁSA

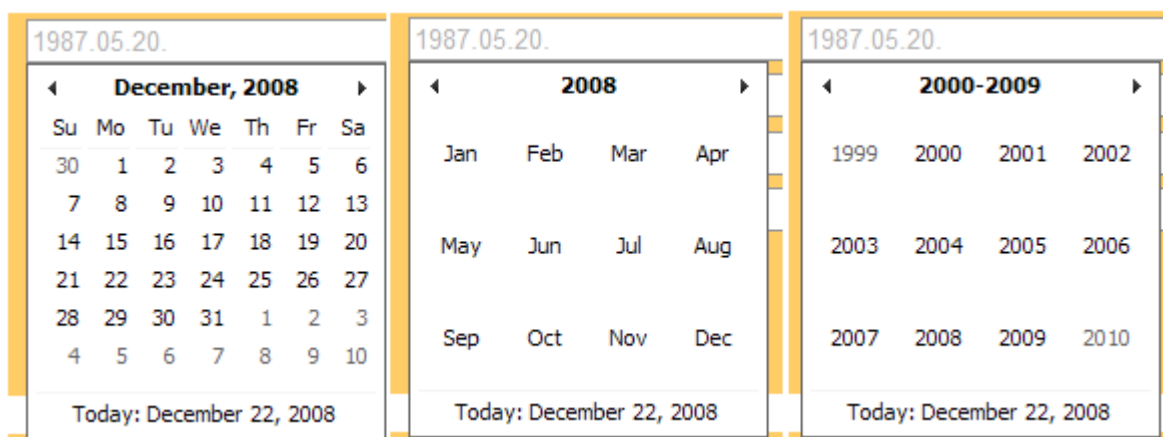
Kattintsunk az „Új” gombra. A gombnyomás hatására a gomb felirata „Mentés”-re változik, és aktívvá válik az adatok bevitelére alkalmas panel.

100. ábra. Felhasználói adatok szerkesztése

Az **E-mail**, **Teljes név**, az **Anyja neve**, illetve a **megjegyzés** mező egyszerű beírással szerkeszthető

Itt megadott e-mail cím fog a későbbiekben felhasználónévként szolgálni, illetve a regisztráció után ide postázzuk a felhasználó jelszavát.

A **Születési Dátum** megadására a mező melletti **naptár ikonra** kattintva van lehetőség. Ebből kattintással kiválasztható a megfelelő dátum. A gyorsabb navigálás érdekében kattintson a hónap nevére, ekkor az adott év hónapjaiból választhat. Ha ekkor az évszámba kattint, akkor évtizedet választhat.



101. ábra. A naptár használata

Az **Intézmény** megadására a mező melletti  gombra kattintva van lehetőség.

Ekkor a megjelenő ablakban az **intézmény nevére kattintva** választhatja az ki. Egy oldalon tíz intézmény jelenik meg. **Lapozásra** a lista alján található számokra kattintva van lehetőség.

Választását a **Kiválasztás** gombra kattintva véglegesítheti, ekkor az intézmény neve megjelenik az előző lapon.

Intézmény	Felelős	Rövidítés	Város	aktív
Intézményen kívüli	Szatmári András	-	-	<input checked="" type="checkbox"/>
A Tan Kapuja Buddhista Főiskola	Farkas Attila Márton	TKBF	Budapest	<input checked="" type="checkbox"/>
Általános Vállalkozási Főiskola	Dr. Szalai Ibolya	ÁVF	Budapest	<input checked="" type="checkbox"/>
Apor Vilmos Katolikus Főiskola	Kádár Gyula	AVKF	Vác	<input checked="" type="checkbox"/>
Budapesti Corvinus Egyetem	Dr. Pénzes Béla	BCE	Budapest	<input checked="" type="checkbox"/>
Budapesti Gazdasági Főiskola	Dr. Harmat Zsigmond	BGF	Budapest	<input checked="" type="checkbox"/>
Budapesti Kommunikációs és Üzleti Főiskola	Dr. Róka Jolán	BKF	Budapest	<input checked="" type="checkbox"/>
Budapesti Műszaki és Gazdaságtudományi Egyetem	Dr. Kovács Ádám	BME	Budapest	<input checked="" type="checkbox"/>
Budapesti Műszaki Főiskola	Dr. Borbély Endre	BMF	Budapest	<input checked="" type="checkbox"/>
Debreceni Egyetem		DE	Debrecen	<input checked="" type="checkbox"/>

1 2 3 4 5 6

Keresés

Kiválasztás

102. ábra. Az intézmény kereső használata

Beosztást a legördülő listából választhat.

A megjegyzés mezőn kívül minden kitöltése kötelező.

A kitöltendő mezők mellett vannak a felsorolva a rendszerben megtalálható szerepkörök. Itt kell szerepkört választania az újonnan létrehozott felhasználónak.

A „Mentés” gombra kattintva mentheti el a változtatásokat.

A „Mégsem” gombra kattintva elvetheti a változásokat.

A sikeres mentésről tájékoztatást kapunk, ellenkező esetben egy figyelmeztetést a hibáról.

FELHASZNÁLÓ ADATAINAK MÓDOSÍTÁSA

Az adatok módosításához válassza ki listából a szerkeszteni kívánt felhasználót, majd kattintson a „Módosítás” gombra.

Ezután az adatlap kitöltése megegyezik a „Az új felhasználó létrehozásánál” leírtakkal.

JELSZÓ ÚJRAKÜLDÉSE

A felhasználók gyakran elveszítik vagy elfelejtik jelszavukat. A „Jelszó újraküldése” gombbal tudunk a felhasználónak új jelszót generálni és e-mailben elküldeni. A folyamat automatikus, csak a gombot kell megnyomni.

A gomb használata előtt válasszuk ki a felhasználók listájából (a nevére kattintva) azt, akinek a jelszót el akarjuk küldeni. Ügyeljünk arra, hogy a felhasználó „aktív” legyen, kitiltott felhasználónak nem küldhetünk új jelszót. Amennyiben a felhasználó nem aktív, a „Módosítás” gombra kattintva változtassuk ezt meg (megj.: egy felhasználó automatikusan kitiltásra kerül, ha túl sok sikertelen bejelentkezést kísérel meg).

„Jelszó újraküldése” gomb megnyomása után a kiválasztott felhasználót egy e-mailt kap az új jelszavával, amivel ezután már be tud lépni.

A sikeres küldésről üzenet tájékoztat.

Ellenkező esetben a hiba jellegét leíró üzenetet kapunk.